2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Web User Profiling using Data Redundancy

Xiaotao Gu*, Hong Yang*, Jie Tang†, Jing Zhang*
Department of Computer Science and Technology, Tsinghua University
{gxt13, yangh13, zhangjing12}@mails.tsinghua.edu.cn*, jietang@tsinghua.edu.cn†

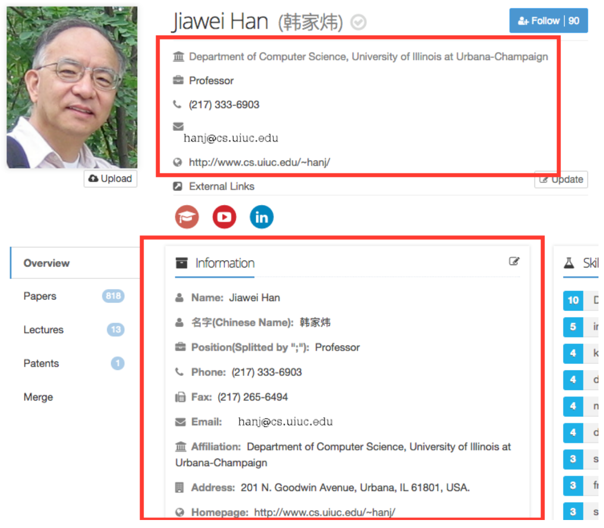
***Abstract*—The study of Web user profiling can be traced back to 30 years ago, with the goal of extracting "semantic"-based user profile attributes from the unstructured Web. Despite slight differences, the general method is to first identify relevant pages of a specific user and then use machine learning models (e.g., CRFs) to extract the profile attributes from the page. However, with the rapid growth of the Web volume, such a method suffers from data redundancy and error propagation between the two steps. In this paper, we revisit the problem of Web user profiling in the big data era, trying to deal with the new challenges. We propose a simple but very effective approach for extracting user profile attributes from the Web using big data. To avoid error propagation, the approach processes all the extraction subtasks in one unified model. To further incorporate human knowledge to improve the extraction performance, we propose a markov logic factor graph (MagicFG) model. The MagicFG model describes human knowledge as first-order logics and combines the logics into the extraction model. Our experiments on a real data set show that the proposed method significantly improves (+4-6%; $p \ll 0.01$, t-test) the extraction performance in comparison with several baseline methods.**

I. Introduction

Web user profiling, also referred to as user profile extraction and user profile mining, has long been viewed as an important and challenging problem in Web mining and natural language processing. Related studies [1], [2], [3], [4], [5], [6] can be traced back to 30 years ago. The general task of web user profiling is to extract "semantic"-based user profile attributes (e.g., contact information, educational history, experience, and biography) from the unstructured Web. Web user profiling can be applied in many applications, and is becoming necessary in most social-related systems. With a large and high-quality profile database, we could easily identify what kind of information we should recommend to a specific user. In e-commerce, one could also leverage the profile information to locate targeted customers for a new product. There are also several products such as, Email Breaker[1], Email Hunter[2], and Sidekick[3] offering services to help users find email addresses of target people from the Web.

Despite much research conducted in this field to automate the process of profile extraction, the problem remains largely unsolved [7], [8], [5]. To generate the profile for a specific user, a usual approach is to first find relevant Web pages of this user from the Web, and then use machine learning models (e.g., CRFs) to extract profile attributes from the pages. State-of-the-art accuracy achieved in the two stages are around 90.0%,

[1]http://emailbreaker.com
[2]https://emailhunter.com
[3]http://www.getsidekick.com



Jiawei Han (韩家炜)

Follow | 90

Department of Computer Science, University of Illinois at Urbana-Champaign
Professor
(217) 333-6903
hanj@cs.uiuc.edu
http://www.cs.uiuc.edu/~hanj/

Information

Name: Jiawei Han
名字(Chinese Name): 韩家炜
Position(Splitted by ";"): Professor
Phone: (217) 333-6903
Fax: (217) 265-6494
Email: hanj@cs.uiuc.edu
Affiliation: Department of Computer Science, University of Illinois at Urbana-Champaign
Address: 201 N. Goodwin Avenue, Urbana, IL 61801, USA.
Homepage: http://www.cs.uiuc.edu/~hanj/

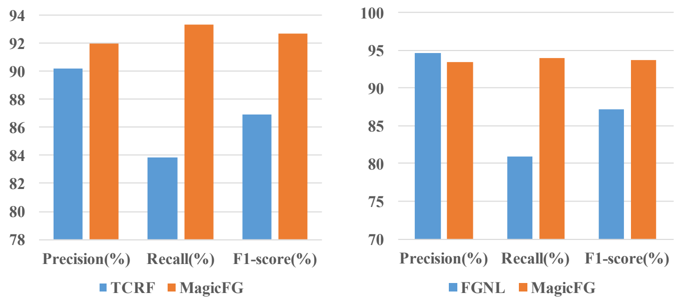
Fig. 1. Example of researcher profile in AMiner.org. The profile contains basic information such as affiliation, position, picture, email, and homepage.

respectively. For example, F1 score is reported as 92% for the task of homepage finding conducted by Tang et al. [9], and when 87% for extracting profile attributes from the homepage [5]. From independent views, the performance sounds good. However, taking error propagation into consideration, the overall accuracy of the approach that combines the two stages together drops to 80%. More seriously, with the rapid growth of the data volume on the Web, the problem becomes even more challenging, as the big Web data contains much more noisy and redundant data.

In this paper, we try to revisit this problem from the perspective of big data. Specifically, to avoid error propagation, we propose a unified approach framework to process all the extraction subtasks together in one step. Moreover, rather than figuring out new ways of reducing noise (or redundancy), we develop a simple but very effective approach for extracting user profile attributes using the redundant information in the big Web data. To incorporate redundant information to improve the extraction accuracy, we propose a markov logic factor graph (MagicFG) model to formalize human knowledge as first-order logics in the model.

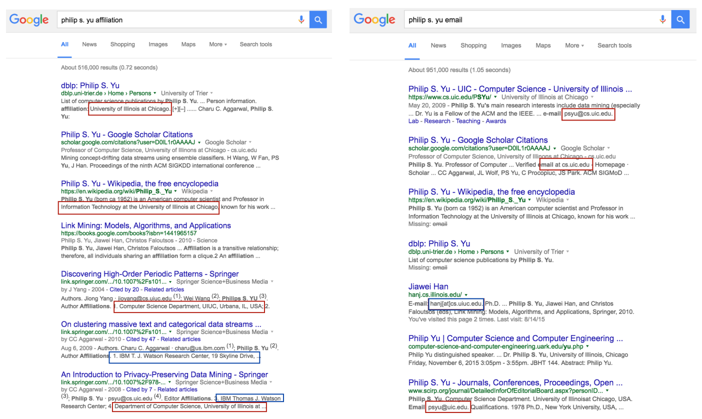
We compare our approach with several state-of-the-art methods for profiling (Cf. Section III for detailed comparisons). As shown in Figure 2, our approach significantly outperforms existing methods for the task of Email extraction and Gender inference in terms of F1-score.

IEEE/ACM ASONAM 2016, August 18-21
2016, San Francisco, CA, USA
978-1-5090-2846-7/16/$31.00 © 2016 IEEE



(a) Email (b) Gender

Fig. 2. Performance comparison between our approach with existing methods. (a) Comparison with TCRF [5], a two-step method for Email extraction. (b) Comparison with FGNL [10] for Gender inference.



philip s. yu affiliation

About 516,000 results (0.72 seconds)

dblp: Philip S. Yu
dblp.uni-trier.de › Home › Persons University of Trier
List of computer science publications by **Philip S. Yu**. ... Person information. **affiliation**: University of Illinois at Chicago. [+][–] ...... Charu C. Aggarwal, **Philip S. Yu**:

Philip S. Yu - Google Scholar Citations
scholar.google.com/citations?user=D0lL1r0AAAAJ Google Scholar
Professor of Computer Science, University of Illinons at Chicago - cs.uic.edu
Mining concept-drifting data streams using ensemble classifiers. H Wang, W Fan, PS **Yu**, J Han. Proceedings of the ninth ACM SIGKDD international conference ...

Philip S. Yu - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Philip_S._Yu** Wikipedia
Philip S. Yu (born ca 1952) is an American computer scientist and Professor in Information Technology at the University of Illinois at Chicago, known for his work ...

Link Mining: Models, Algorithms, and Applications
https://books.google.com/books?isbn=1441965157
Philip S. Yu, Jiawei Han, Christos Faloutsos - 2010 - Science
Philip S. Yu, Jiawei Han, Christos Faloutsos ... **Affiliation** is a transitive relationship; therefore, all individuals sharing an **affiliation** form a clique.2 An **affiliation** ...

Discovering High-Order Periodic Patterns - Springer
link.springer.com/.../10.1007%2Fs101... Springer Science+Business Media
by J Yang - 2004 - Cited by 20 - Related articles
Authors. Jiong Yang · jioyang@cs.uiuc.edu (1); Wei Wang (2); **Philips S. YU** (3). Author **Affiliations**. 1. Computer Science Department, UIUC, Urbana, IL, USA; 2.

On clustering massive text and categorical data streams ...
link.springer.com/.../10.1007%2Fs101... Springer Science+Business Media
by CC Aggarwal - 2010 - Cited by 47 - Related articles
Aug 6, 2009 - Authors. Charu C. Aggarwal · charu@us.ibm.com (1); **Philip S. Yu** (2). Author **Affiliations**. 1. IBM T. J. Watson Research Center, 19 Skyline Drive, ...

An Introduction to Privacy-Preserving Data Mining - Springer
link.springer.com/.../10.1007%2F978-... Springer Science+Business Media
by CC Aggarwal - 2008 - Cited by 7 - Related articles
(3); **Philip S. Yu** · psyu@cs.uic.edu (4). Editor **Affiliations**. 3. IBM Thomas J. Watson Research Center; 4. Department of Computer Science, University of Illinois at ...

(a) Affiliation

philip s. yu email

About 951,000 results (1.05 seconds)

Philip S. Yu - UIC - Computer Science - University of Illinois ...
https://www.cs.uic.edu/**PSYu**/ University of Illinois at Chicago
May 20, 2009 - **Philip S. Yu's** main research interests include data mining (especially ... Dr. Yu is a Fellow of the ACM and the IEEE. ... **e-mail** psyu@cs.uic.edu.
Lab - Research - Teaching - Awards

Philip S. Yu - Google Scholar Citations
scholar.google.com/citations?user=D0lL1r0AAAAJ Google Scholar
Professor of Computer Science, University of Illinons at Chicago - cs.uic.edu
Philip S. Yu. Professor of Computer ... Verified **email** at cs.uic.edu · Homepage · Scholar ... CC Aggarwal, JL Wolf, PS Yu, C Procopiuc, JS Park. ACM SIGMoD ...

Philip S. Yu - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Philip_S._Yu** Wikipedia
Philip S. Yu (born ca 1952) is an American computer scientist and Professor in Information Technology at the University of Illinois at Chicago, known for his work ...
Missing: ~~email~~

dblp: Philip S. Yu
dblp.uni-trier.de › Home › Persons University of Trier
List of computer science publications by **Philip S. Yu**.
Missing: ~~email~~

Jiawei Han
hanj.cs.illinois.edu/
E-mail: hanj[at]cs.uiuc.edu. Ph.D. ... **Philip S. Yu**, Jiawei Han, and Christos Faloutsos (eds), Link Mining: Models, Algorithms, and Applications, Springer, 2010.
You've visited this page 2 times. Last visit: 8/14/15

Philip Yu | Computer Science and Computer Engineering ...
computer-science-and-computer-engineering.uark.edu/yu.php
Philip Yu distinguished speaker. ... Dr. **Philip S. Yu**, University of Illinois, Chicago Friday, November 6, 2015 3:05pm - 3:55pm. JBHT 144. Abstract: Philip Yu.

Philip S. Yu - Journals, Conferences, Proceedings, Open ...
www.scirp.org/journal/DetailedInforOfEditorialBoard.aspx?personID...
Philip S. Yu. Computer Science Department. University of Illinoisat Chicago, USA. **Email** psyu@uic.edu. Qualifications. 1978 Ph.D., New York University, USA, ...

(b) Email

Fig. 3. Snippets returned by Google by the two constructed queries. From (a) we can easily identify two affiliations and from (b) we can also identify two email addresses.

We apply the proposed model to an academic search and mining system AMiner.org[4] to extract profile for researchers from the Web. Figure Figure 1 shows an example of researcher profile in AMiner.org. The profile contains basic information of Dr. Jiawei Han such as affiliation, position, picture, email, and homepage. With the proposed model, we have successfully extracted more than 100,000,000 researcher profiles. We also quantitatively evaluate the proposed model on a ground-truth dataset. The proposed method achieves significant improvement (+4-6% in terms of F1 score; $p \ll 0.01$, t-test) over several alternative methods.

Organization In Section II, we describe the proposed approach for user profiling. In Section III, we present experimental results to evaluate the effectiveness of the proposed approach. In Section IV, we review the related work and finally, Section V concludes the paper.

II. Approach Framework

In this section, we first give the basic idea of the proposed framework to solve the profiling problem, and then introduce three methods to extract profile attributes from the Web.

A. Basic Idea

Given a person v, referred to as query person, our goal is to extract profile attributes of the person and construct a researcher profile. For example, in the academic search system, the researcher profile consists of position, picture, address, phone, email, homepage, research interest, etc. A detailed definition can be found in [5]. We aim to design a general method to automatically extract the profile attributes from the Web with high accuracy. The method should be also flexible enough to extended to handle new profile attributes.

Traditional methods usually deal with the problem by first finding relevant Web pages for the query person from the Web, and then using models such as SVM or CRF to extract the required profile attributes from the pages. In both steps, the state-of-the-art performances achieved by traditional methods are around 90% [5], [9]. However, the overall accuracy by combining the two steps inevitably drops to 80% due to error propagation between the two steps. Meanwhile, the required profile attributes may distribute in different Web pages, which leads to two new challenges: 1) extraction from distributed pages and 2) extraction with data redundancy.

To tackle the problem of error propagation and data redundancy, we propose a unified framework to process all the extraction subtasks together from the big Web data. The approach is simple but very effective. Specifically, for each profile attribute, we first construct a "smart" query and use a search engine to retrieve relevant *snippets* with the query, finally an extraction model is applied to the returned snippets to extract the profile attribute. The idea behind is to leverage data redundancy to help the extraction. Suppose we are going to extract the affiliation of "Philip S. Yu". The constructed query can be "Philip S. Yu affiliation". In a similar way, to extract the email address of "Philip S. Yu", we can construct "Philip S. Yu email". Figure 3 shows two example snippets returned by Google with two constructed queries. We see that from 3(a) we can easily identify two different affiliations: "University of Illinois at Chicago" and "IBM T. J. Watson Research Center" (after normalization), and from 3(b) we can also identify two email addresses: "psyu@cs.uic.edu" and "hanj[at]cs.uiuc.edu". We call the identified affiliations/emails from the snippets as candidate affiliations/emails. Now the problem is how to rank the identified information. Our basic idea is to leverage the redundancy information—e.g., "University of Illinois at Chicago" occurs four times in the snippets and two times for "IBM T. J. Watson Research Center". More precisely, we propose a MAkov loGIC factor graph (MagicFG) to rank the obtained candidates by leveraging the redundancy information. The model is flexible and can easily incorporate any domain human knowledge to further improve the extraction accuracy.

It is noteworthy that we are restricted to the two example profile attributes. In fact, the proposed method is in general flexible. To extract a new profile attribute, what we need to do is to construct the "smart" query and to train the MagicFG

[4]https://aminer.org

model. For some profile attributes, it is easy to construct the query. For example, we found that for email, we can achieve a high accuracy by simply using name+email. For some other profile attributes, for example, Gender and Position, the situation may be more complicated. We will introduce how we construct the smart query for general profile attributes in Section II-B. Please also note that there are generally two types of profile attributes: the categorical attributes and the non-categorical attributes. For example, Gender is a categorical attribute. Position is also a categorical attribute with multiple values, such as professor, student, researcher, and engineer. While Email and Age are two non-categorical attributes. The two kinds of attributes will be treated slightly differently in the proposed framework.

B. Smart Query Construction

We construct the queries for the categorical and non-categorical attributes in different ways. For the categorical attributes, we construct the query by automatically identifying representative keywords in each candidate category and combine them together as the query. To find the representative keywords for each category, we first collect several person names (e.g., 1000) for each category from professional websites such as AMiner and LinkedIn. We then submit the corresponding person names as queries to search engines like Google to obtain top-k (e.g., 10) snippets. Among all the words in the snippets, we identify the most representative keyword as that with the highest TF-IDF scores [11]. The TF-IDF score of a word w in a category c is calculated as follows:

$$\text{TF-IDF}(w,c) = (1+\log n(S_c,w))\log(1+\frac{|S|}{n(S,w)}) \quad (1)$$

where S_c denotes the snippets that belongs to category c. Notation $n(S_c,w)$ denotes the number of snippets in category c that contains the word w. Notation $n(S,w)$ indicates the number of snippets in all the categories that contains the word w and $|S|$ is the number of all the snippets in all the categories.

Take Gender as an example, using the above method, we found that the most representative keyword is "her" for females, and is "his" for males . The query is then constructed as "name his|her".

For a non-categorical attribute, we directly use the keywords in the attribute name to construct the query. For example, the query for Email extraction can be "name email|mail".

C. Baseline Extraction Models

We first introduce two baseline models for extracting the profile attributes.

1) Rule-based Model: In the rule-based model, for extracting Email of the query person v, we simply construct the query by combing the person name and word "email". Once obtained the returned snippets from a search engine (e.g., Google), we can use rule-based heuristics to extract the Email of the query person. The rules are defined as follows: we first extract the candidate email addresses from the searched snippets, and if we find the first name or the last name of the queried person name is contained in the prefix of a candidate email address, then the extracted Email will be selected as the result. One thing worth noting is the recognition of the email address, because in many Web pages, especially some person's homepages, the email addresses may be encoded in special patterns such as "firstname [dot] lastname [at] cmu [dot] edu". Heuristic rules are defined to recognize potential email addresses[5].

Our preliminary experiments show that such a simple method could be able to result in an accuracy of 88%—comparable with the state-of-the-art performance obtained by traditional two-step approach (Cf. Section III for detailed comparisons).

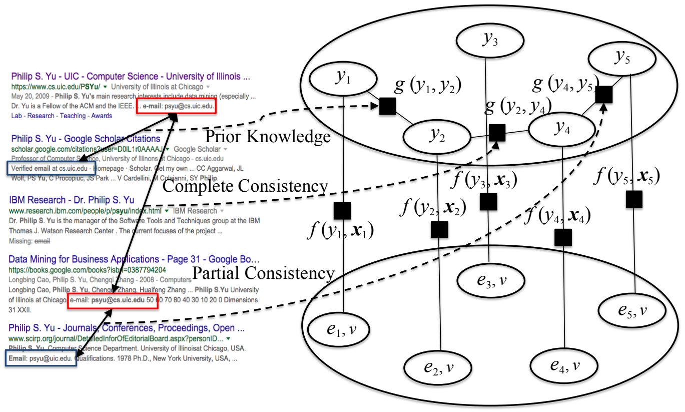
2) Classification-based Model: We make use of Logistic Regression (LR) as the classification model. Let us first consider a two class classification problem. Let $\{(x_1,y_1),\cdots,(x_N,y_N)\}$ be a training data set, in which x_i denotes a feature vector of a candidate information and $y_i \in \{-1,+1\}$ denotes a classification label (whether the candidate is correct or not). The classification-based extraction model consists of two stages: learning and extraction. In learning, one attempts to find an optimal weight configuration to maximize the log-likelihood function of the observed instances). In extraction, we use the learned model to classify which candidate information is we want to extract.

Regarding features in the classification model, we use the same attribute features as the attribute factors defined in our proposed model (Cf. Section II-C for details). The classification can adjust the weights of different features and combine the feature together, thus obtains a better performance (90% in terms of F1-score) than the rule-based method. However, as shown in Figure 3, the returned snippets usually contain much redundant information that might be helpful for the extraction. Both the rule-based and the classification-based models consider each candidate instance as independent and thus cannot leverage such redundant information.

D. Markov Logic Factor Graph (MagicFG) Model

The rule-based method is comparable with traditional methods and the classification-based method outperforms many existing methods. Both of the above methods treat the candidate email addresses as independent classification objects, while both the rule-based and the classification-based methods ignore the correlations among the candidate instances identified in the returned snippets. However, this redundant information can be leveraged. In practice, the redundant information resided in the snippets can be helpful to improve the extraction accuracy. For example, in Figure 3(b), for Email extraction, the same prefix "psyu" before "@" in the two candidate email addresses, "psyu@cs.uic.edu" and "psyu@uic.edu", indicates that the two email addresses belong to the same person.

[5]One example of the heuristic rule: "$(([a-z0-9]+)(\backslash.|dot|\backslash.)?)+(@|at|\backslash[at\backslash]|\backslash[at\backslash])(([a-z0-9\backslash]+)(\backslash.|dot|\backslash.\backslash[dot\backslash]))+([a-z]+)$"



Philip S. Yu - UIC - Computer Science - University of Illinois ...
https://www.cs.uic.edu/PSYu/ ▾ University of Illinois at Chicago ▾
May 20, 2009 - **Philip S. Yu's** main research interests include data mining (especially ...
Dr. Yu is a Fellow of the ACM and the IEEE. ... **e-mail:** psyu@cs.uic.edu.
Lab - Research - Teaching - Awards

Philip S. Yu - Google Scholar Citations
scholar.google.com/citations?user=D0IL1r0AAAAJ ▾ Google Scholar ▾
Professor of Computer Science, University of Illinons at Chicago - cs.uic.edu
Verified **email** at cs.uic.edu - Homepage · Scholar. Get my own ... CC Aggarwal, JL Wolf, PS **Yu**, C Procopiuc, JS Park ... V Cardellini, M Colajanni, SY **Philip**.

IBM Research - Dr. Philip S. Yu
www.research.ibm.com/people/p/p**syu**/index.html ▾ IBM Research ▾
Dr. **Philip S. Yu** is the manager of the Software Tools and Techniques group at the IBM Thomas J. Watson Research Center . The current focuses of the project ...
Missing: ~~email~~

Data Mining for Business Applications - Page 31 - Google Bo...
https://books.google.com/books?isbn=0387794204
Longbing Cao, Philip S. Yu, Chengqi Zhang - 2008 - Computers
Longbing Cao, **Philip S. Yu**, Chengqi Zhang, Huaifeng Zhang ... **Philip S.Yu** University of Illinois at Chicago e-mail: **psyu@cs.uic.edu** 50 60 70 80 40 30 10 20 0 Dimensions 31 XXII.

Philip S. Yu - Journals, Conferences, Proceedings, Open ...
www.scirp.org/journal/DetailedInforOfEditorialBoard.aspx?personID... ▾
Philip S. Yu Computer Science Department. University of Illinoisat Chicago, USA.
Email: psyu@uic.edu. Qualifications. 1978 Ph.D., New York University, USA, ...

Fig. 4. Graphical representation of logic factor graph model based on a real search example. Notation (e_i, v) represents a Email-person pair, and y_i indicates its corresponding label; Notations $f(.)$ and $g(.)$ represent the attribute factor function and logic factor function respectively.

Now, the problem becomes how to encode and incorporate such kind of redundancy-based correlation into a unified extraction model. we propose a novel Markov logic factor graph (MagicFG) model to model the correlations as first-order logics and to leverage logics to improve the extraction performance. We introduce how to model the data redundancy for the non-categorical and categorical attributes respectively.

Modeling Non-categorical Attributes. When extracting non-categorical attributes, for each query person, we construct a factor graph model with each node representing a candidate instances, and each edge corresponding to the dependency between two candidate instances. We optimize the factor graph model for all query persons simultaneously. We explain the modeling process of categorical attributes using Email as the example. For a query person v, we denote each candidate Email as e_i. As the example in Figure 4, we could extract four candidates $\{e_1, e_2, e_3, e_4\}$. For each candidate Email, we create an instance (e_i, v) and associate with a latent variable y_i. To model the correlations between the candidate instances, we can construct a factor graph model as presented in Figure 4. The model is referred to as Markov logic factor graph (MagicFG) model. In MagicFG, the correlation is represented as the first-order logic as it defines prior knowledge and all the other correlations as first-order logics. We will explain how we define the first-logic based correlation later. At the high-level, in MagicFG, we define two types of functions.

- **Attribute factor function:** It captures characteristics of the Email-person pair and is defined as an exponential function:

$$f(v, e_i, y_i) = \frac{1}{Z_a} \exp\{\sum_k \alpha_k \phi_k(y_i, \boldsymbol{x_i})\}, \qquad (2)$$

  where $\phi_k(.)$ is the k^{th} feature function defined between v and e_i with respect to the value of y_i; α_k is the weight of the corresponding attribute feature; $\boldsymbol{x_i}$ is the i^{th} feature vector. Z_a is the normalization factor.

- **Logic factor function:** It captures the correlations between latent variables. It is also defined as an exponential function:

$$g(y_i, y_j) = \frac{1}{Z_b} \exp\{\sum_m \beta_m \psi_m(y_i, y_j)\}, \qquad (3)$$

  where $\psi_m(.)$ is the m^{th} correlation factor function defined between y_i and y_j according to a first-order logic knowledge base; β_m is the weight of the corresponding correlation factor .

For the attribute factor function, we can define multiple feature functions $\{\phi_k(y_i, \boldsymbol{x_i})\}_k$ to characterize each candidate instance. For extracting Email, we define features such as whether v's first name, last name or full name is contained in e_i's prefix.[6] Another kind of feature is defined between person v and the context c_i from which the candidate e_i is extracted. For example, whether v's first name, last name or full name is contained in context c_i, and whether v's affiliation is contained in context c_i. Here we use the affiliation information to disambiguate persons with the same names.

Regarding the logic factor function, we mainly consider three kinds of first-order logic relationships between latent variables: complete consistency, partial consistency and prior knowledge. First-order logic is the standard for the formalization of mathematics into axioms and is studied in the foundations of mathematics. In our problem, we use first-order logic to encode user-specific correlations between candidate instances and domain human knowledge about the extraction. For a general introduction of first-order logic, please refer to [12].

Complete consistency describes the fact that the values of two latent variables y_i and y_j should be consistent under some given conditions. For example, the following first-order logic

$$\text{Equals}(e_i, e_j) \Rightarrow \text{Equals}(y_i, y_j)$$

indicates that y_i equals y_j if the corresponding Email candidates are same with each other. The logic is straightforward because two same email addresses are highly likely to be credible or incredible at the same time. Correspondingly, we define the factor function as

$$\psi(y_i, y_j) = \begin{cases} 1, & e_i = e_j \text{ and } y_i = y_j \\ 0. & \textit{otherwise} \end{cases}$$

Partial consistency describes the fact that the values of two latent variables y_i and y_j should be partially consistent under some given conditions. For example, the following first-order logic

$$\text{SamePrefix}(e_i, e_j) \Rightarrow \text{True}(y_i) \wedge \text{True}(y_j)$$

[6]We call the string before "@" of an Email candidate as the prefix of the Email, and the string after "@" as the domain of it.

TABLE I
FIRST-ORDER LOGIC KNOWLEDGE BASE.

First-order logic	Example
Complete Consistency	Equals$(e_i, e_j) \Rightarrow$ Equals(y_i, y_j)
Partial Consistency	SamePrefix$(e_i, e_j) \Rightarrow$ True$(y_i) \wedge$ True(y_j)
Prior Knowledge	IsBlocked$(e_j) \wedge$ SameDomain(e_i, e_j) $\Rightarrow$ True$(y_i) \wedge$ False(y_j)

indicates that y_i and y_j both equal to 1 if their prefixes are the same. This logic can be explained as follows. When two email addresses share the same prefix, they are very likely to mention the same person, because people usually use the same prefix in different email addresses. In this case, If one email address is correct, the other one is also likely to be also correct. We define the corresponding factor function as

$$\psi(y_i, y_j) = \begin{cases} 1, & e_i \text{ and } e_j \text{ have the same prefix} \\ & \text{and } y_i = y_j = 1 \\ 0. & otherwise \end{cases}$$

Prior knowledge describes the prior knowledge that can be formalized into useful first-order logics for a specific task. For example, when we search for someone's email address by Google, we find that many candidates starting with the word "email", like "email@gmail.com". This is due to the security policy of the search engine, which blocks or modifies the prefixes of some email addresses from several sensitive sources. We can still observe the domain information. We found that when another candidate shares the same domain with a blocked candidate, it is more likely that the other candidate is a correct Email. We define the corresponding first-order logic as

$$\text{IsBlocked}(e_j) \wedge \text{SameDomain}(e_i, e_j) \\ \Rightarrow \text{True}(y_i) \wedge \text{False}(y_j).$$

The corresponding factor function is defined as

$$\psi(y_i, y_j) = \begin{cases} 1, & c_j \text{ is blocked,} \\ & c_i \text{ and } c_j \text{ have the same domain,} \\ & y_i = 1 \text{ and } y_j = 0 \\ 0. & otherwise \end{cases}$$

For each profiling task, we build a knowledge base according to the defined first-order logics and summarize it in Table I. In general, the attribute factors capture the characteristics on each potential person-Email pair and the logic correlation factors capture the dependencies between two person-Email pairs.

Modeling Categorical Attributes. When dealing with categorical attributes, for all the queried persons, we build one factor graph with each node representing a query person, and the edges representing dependencies between two query persons. We use Gender as the example to explain the modeling process for the categorical attributes.

Different from the non-categorical attributes, each person can only have one Gender, either male or female. Thus in this task, we directly assign a label to each query person. We construct a query by combing the person name and the representative keywords for each Gender("his" for male and "her" for female, as mentioned before). The query finally looks like "name his|her". Then we formulate the MagicFG based on the returned snippets.

The formulation of MagicFG model is also a little different from that of non-categorical attributes. We feed the model with each observation variable as a person v_i. The corresponding latent variable y_i to each person v_i represents v_i's attribute values, e.g., whether v_i is male or female.

For attribute factor functions, we first extract features for each person from his/her search context. For example, whether a snippet in the search results contains both the person name and the word "his/her", whether a snippet contains both the affiliation and the word "his/her", whether "his/her" appears in the snippets of the top 3 returned search results, and the number "his/her" in all the search results. For logic factor functions, we define a correlation feature of the type of complete consistency logic as follows:

$$\text{SameFirstname}(v_i, v_j) \Rightarrow \text{Equals}(y_i, y_j).$$

The logic indicates that the Gender of two persons are more likely to be the same if they have the same first name.

In summary, the factor graphs built for the non-categorical and categorical attributes are slightly different. For non-categorical attributes, multiple graphs are build, of which each graph is build for each person with each candidate attribute being formed as a node and the dependency between two candidate attributes being formed as an edge. While for categorical attributes, only one graph is build, with each person being formed as a node and the dependency between two persons being formed as an edge.

Model Training and Extraction. Once we formulated the MagicFG model for either non-categorical or categorical attributes, we can combine the defined factor functions and define the following log-likelihood objective function by following the Markov assumption [13]:

$$\log P(Y|X, \theta) = \sum_{y_i \in Y} \sum_{k} \alpha_k \phi_k(y_i, \boldsymbol{x_i}) \\ + \sum_{e_i \sim e_j} \sum_{m} \beta_m \psi_m(y_i, y_j) - \log Z, \tag{4}$$

where $Z = Z_a Z_b$ is the normalization factor; $e_i \sim e_j$ indicates that there is a (directed or indirected) correlation between e_i and e_j; $\theta = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ are parameters to estimate, representing the weights of the defined feature functions.

Training a MagicFG is to estimate a parameter configuration $\theta = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ from a given historical dataset, such that the log-likelihood objective function $L(\theta)$ can be maximized,

$$\theta^* = \arg\max_{\theta} \log P(Y|X, \theta). \tag{5}$$

We use a gradient ascent algorithm to solve the objective function. The gradient for parameter α_k can be written as:

$$\frac{\partial L(\theta)}{\partial \alpha_k} = \mathbb{E}[\phi(y_i, \boldsymbol{x_i})] - \mathbb{E}_P(y_i, \boldsymbol{x_i})[\phi(y_i, \boldsymbol{x_i})]. \quad (6)$$

The parameter β_m can be obtained in the same way. In the above equation, the first term $\mathbb{E}[\phi(y_i, \boldsymbol{x_i})]$, representing the expectation of features values under the uniform distribution, can be easily calculated, while it is usually intractable to directly estimate the marginal probability in the second term as the graphical structure can be arbitrary and may contain cycles. In this work, we use loopy belief propagation (LBP) [14] to approximate the marginal probability in the second term and accordingly calculate the gradient. The learning algorithm can be divided into two steps: we first perform the LBP algorithm to calculate marginal distribution for each latent variable, and then update each parameter to maximize the objective log-likelihood function by :

$$\theta_{new} = \theta_{old} + \eta \cdot \frac{\mathcal{O}(\theta)}{\theta}, \quad (7)$$

where η is the learning step. The process repeats updating marginal probabilities and parameters until the convergence or until the number of iterations is large enough.

Given the observed feature vectors $\boldsymbol{X}_v$ for all candidates of person v and the learned parameter configuration θ, the extraction can be done by finding the most likely configuration of $Y_v = \{y_1, ..., y_I\}$ for all the person-Email pairs $\{(e_i, v)\}$:

$$Y_v = \arg\max_{Y_v} P(Y_v|X_v, \theta), \quad (8)$$

where the LBP algorithm is again used to solve this problem.

Discussions. Different from traditional methods that crawled each of the relevant pages, we only use the snippet information to extract the profile attributes. It is much faster and more stable, as different servers that host the relevant pages may have very different network speed. Also we found with the constructed "smart" queries, more than 90% of the profile attributes are already contained in the snippets returned by the search engine. One additional advantage is that we do not need to maintain a large database to record all the relevant pages for all the query persons. This is very important, as, for example, in AMiner, we have more than 130,000,000 researchers—maintaining such a big database for all researchers itself is a challenging task. Moreover, the profile information is very dynamic. Our method avoids this problem by directly querying the search engine.

III. Experimental Results

In this section, we demonstrate the effectiveness of our approach for both categorical and non-categorical attributes. For quantitative evaluation, we take Gender as an example of categorical attributes, and Email as an example of non-categorical ones. Please note that our framework is very flexible and have already been applied to an online academic search and mining system AMiner.org to extract the profiles for researchers. All datasets and codes used in this work are publicly available.[7]

A. Experiment Setup

Dataset. To construct a ground-truth dataset for quantitative evaluation, we randomly choose 2,000 researchers from AMiner.org [9]. Specifically, for extracting Email of each researcher, we search the Web using search engine by querying the person name and the word "email". This way results in 4,528 Email candidates. Human annotations are applied to identify the correct Email addresses. In an analogous way, for inferring Gender, we search the Web by querying the person name and the word "his" or "her". Human annotations were also conducted to identify the Gender of these 2,000 researchers. For disagreements in the annotation, we conducted "majority voting". Finally, for the 2,000 researchers, we identify 34% of the researchers are female researchers; and about 40% of the Email candidates are correct Email, which means that our framework can find the correct Email for over 90% users.

Evaluation Metrics. To quantitatively evaluate our model, we divide the dataset into training set and test set. We perform five-fold cross-validation and report the extraction performance in terms of precision, recall, and F1-score.

Comparison Methods. We compare the MagicFG with following methods for extracting Email and Gender on the ground-truth dataset.

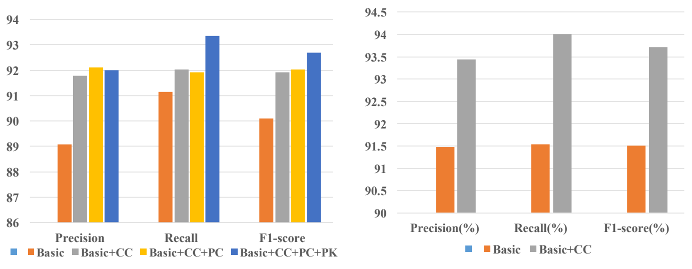
- **Rule.** Uses several simple defined rules to extract profile attributes. For example, for extracting Gender, we count the number of common names for girls and boys. For extracting Emails, we find whether the prefix of the Email contains the person name.
- **Support Vector Machine (SVM).** Uses the same attribute factors as features and employs SVM-Light [15] to train and predict Email and Gender.
- **Random Forest (RF).** Uses the same attribute factors as features and employs sklearn package to conduct train and predict.
- **Logistic Regression (LR).** Uses the same attribute factors as features and employs sklearn package to conduct train and predict.

The MagicFG model is implemented in C++. All experiments are conducted on a Macbook Pro with Intel Core i5 CPU 2.9GHz(2 cores) and 8 GB memory. In all the experiments, we set $L = 10$ and search top 10 results by Google search, and conduct a five-fold cross validation for each method.

B. Extraction Performance

Email extraction. Under the same framework we propose, the MagicFG model outperforms the best extraction method, namely RF (+2.12% in terms of F1-score). This is because

[7]https://aminer.org/profiling/



(a) Email (b) Gender

Fig. 5. Effect of logic correlation factors in Email extraction and Gender inference. Basic stands for the MagicFG model that only consider the attribute factors. +CC stands for adding the factors of complete consistency logic. +PC adds factors of partial consistency logic. +PK adds the factors of prior knowledge logic.

MagicFG captures the dependencies between different Email candidates by incorporating them as features of first-order logics, in addition to the independent attribute features.

Gender inference. Under the same framework we propose, the MagicFG model outperforms the best method LR (+2.21% in terms of F1-score). Because MagicFG additionally incorporates one logic relationship of complete consistence logic and capture the dependencies between different person candidates.

Effect of factors. We further present an in-depth analysis of how different logic correlation factors affect the performance of user profiling. Figure 5 shows the different evaluation metrics of the proposed MagicFG by considering different levels of logic factors. It can be clearly seen from Figure 5(a) that for Email extraction, the accuracy performance drops significantly without the logic correlations. In addition, adding the factors of prior knowledge logic can further improve the performance significantly. Figure 5(b) also show that the factor of complete consistence logic improves the performance of Gender inference significantly.

C. Comparison with Existing Methods

We now compare our approach with several state-of-the-art methods for the task of Email extraction and Gender inference:

- **TCRF**

  For Email extraction, we use the method proposed in [5] as the baseline (to hereafter refer to as: TCRF), which is one of the state of art approaches to extract homepages and Emails from the Web. This method has two steps, where it first finds the user's homepage and then extracts Email from the homepage with a high precision using an extraction model named TCRF.

  Table II shows the classification performance of Email extraction by different methods. We can see from the results that our method consistently outperforms the baseline (TCRF) on F1-score by +5.78%. As you can see, the recall of our system is clearly much better (+9.53%). This is because the TCRF method only chooses the homepage as its data source, which is a little narrow and ignores useful information from other sources on the Web. From the careful construction of query, our approach can effectively find out rich sources related to the target attribute, reducing the risk of missing the right choice. It is noteworthy that our approach also achieves better precision for Email extraction.

- **Facebook Generated Name List Predictor(FGNL).**

  For Gender inference, we use a method proposed by [10] as the baseline (to hereafter refer to as: FGNL). Most state of the art methods for inferring Gender depend on a list of common names for males and females. In [10], the authors proposed an approach which used data from Facebook to construct an expanded and high-quality name list. They match the user's first name with the list to make the inference. If the first name is matched with one of the male names, the user is treated as a male, and vice versa. While if the first name is found in neither the male names nor the female names, or in both the name lists, they make a random guess about the user's Gender.

  Table III shows the classification performance of Gender inference by different methods. We can see that our method outperforms the baseline (FGNL) on F1-score by +6.49%. Our method performs much better than the FGNL method in recall (+13.13%). This is because the FGNL method depends greatly on the name list. However, you can never list all those names, no matter how large the list is. On the contrary, our approach can automatically find the representative keywords for documents describing a user with specific Gender, and infer Gender from the big Web data with less limitation. So we seldom have the problem that the FGNL has to face when they cannot find the name in their list. As the table shows, the FGNL method performs slightly better in precision (+1.22%), which is clearly an advantage of using the name list. However, our approach achieves a

TABLE II
PERFORMANCE COMPARISON OF EMAIL EXTRACTION (%)

Method	Precision	Recall	F1-score
TCRF	90.20	83.83	86.90
Rule	87.81	89.64	88.72
SVM	88.26	89.25	88.75
RF	90.76	90.58	90.56
LR	89.07	91.14	90.11
MagicFG	**92.00**	**93.36**	**92.68**

TABLE III
PERFORMANCE COMPARISON OF GENDER INFERENCE (%)

Method	Precision	Recall	F1-score
FGNL	**94.66**	80.88	87.23
Rule	92.12	88.32	90.18
SVM	91.98	90.60	91.29
RF	90.17	89.99	90.08
LR	91.48	91.54	91.51
MagicFG	93.44	**94.01**	**93.72**

close precision while raising the recall to a different level. Taking the limits of the FGNL method into consideration, our approach is much more generalizable.

IV. Related work

Previously considerable efforts have been made for obtaining user profiles. Back in 1900s, [2] discussed algorithms for learning and revising user profiles that could determine which Web sites on a given topic would be interesting to a user. It used a Naive Bayes classifier to incrementally learn profiles from user feedback on the Web sites. [16] had developed a personalized web browser. It learned a user profile, and aimed at helping user navigate the Web by searching for potentially interesting pages for recommendations. [3] described an experimental work to study whether user interests could be automatically classified through heuristics. The results highlighted the need for user feedbacks and machine learning methods.

Nowadays, with the rapid development of the Internet, especially that of social networks, we are capable of fetching user profiles, with different methods and from different sources. For example, Yu et al. propose a cascaded information extraction framework for identifying personal information from resumes [17]. Tang et al. propose a conditional random field to extract user profiles from one's homepages [5]. Li et al. propose a weakly supervised method to extract user profiles from Twitter in 2014 [6]. Merler et al. propose a method to extract user attributes from the pictures posted in social media feeds [18], especially gender information. [19] and inferred user's profile by analysing the user's Twitter posts, which is a little difficult to generalize to other applications. However, these methods are highly dependent on the quanlity of the data sources, and thus their performance may be sufferred from error propagation.

To reduce such risks, efforts have been made to combine several data sources. In 2015, [20] presented an initial study of user profile learning via integration of multiple data sources, including Twitter, Foursquare and Instagram. They present that multiple data sources of the same users can enhance the final performance. [21] proposed an effective method to link different social network accounts for a specific user. The correlation between redundancy and correctness of retrieved information has also been well studied in [22] and [23]. These interesting ideas give us a different insight into the profiling problem, and inspire us to design a more generalizable framework.

V. Conclusion

We study an interesting problem of web user profiling using big data and propose an approach framework to extract user profile attributes directly from the Web. For a given profiling task, the approach first constructs a meaningful query to retrieve relevant information. Without downloading any Web data, we present a Markov logic factor graph (MagicFG) model to directly model and extract user profile from the search results. The MagicFG incorporates the redundant information in the big data. We test the proposed method on two real data sets. Our experiments show that the proposed method significantly improves the profiling accuracy in comparison with several comparison methods.

References

[1] G. Brajnik, G. Guida, and C. Tasso, "User modeling in intelligent information retrieval," *Information Processing & Management*, vol. 23, no. 4, pp. 305–320, 1987.

[2] M. J. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites," *Machine Learning*, vol. 27, no. 3, pp. 313–331, 1997.

[3] S. J. Soltysiak and I. B. Crabtree, "Automatic learning of user profiles — towards the personalisation of agent services," *BT Technology Journal*, vol. 16, no. 3, pp. 110–117, 1998.

[4] J. Tang, D. Zhang, and L. Yao, "Social network extraction of academic researchers," in *ICDM'07*, 2007, pp. 292–301.

[5] J. Tang, L. Yao, D. Zhang, and J. Zhang, "A combination approach to web user profiling," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 5, no. 1, p. 2, 2010.

[6] J. Li, A. Ritter, and E. H. Hovy, "Weakly supervised user profile extraction from twitter." in *ACL (1)*, 2014, pp. 165–174.

[7] B. Krulwich, "Lifestyle finder: Intelligent user profiling using large-scale demographic data," *AI magazine*, vol. 18, no. 2, p. 37, 1997.

[8] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, "Ontological user profiling in recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 54–88, 2004.

[9] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 990–998.

[10] C. Tang, K. Ross, N. Saxena, and R. Chen, "Whats in a name: a study of names, gender inference, and gender behavior in facebook," in *Database Systems for Adanced Applications*. Springer, 2011, pp. 344–356.

[11] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. ACM Press, 1999.

[12] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, no. 1-2, pp. 107–136, 2006.

[13] J. M. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," 1971.

[14] J. S. Yedidia, W. T. Freeman, Y. Weiss *et al.*, "Generalized belief propagation," in *NIPS*, vol. 13, 2000, pp. 689–695.

[15] T. Joachims, "Making large scale svm learning practical," Universität Dortmund, Tech. Rep., 1999.

[16] P. K. Chan, "A non-invasive learning approach to building web user profiles," in *KDD-99 Workshop on Web Usage Analysis and User Profiling*, 1999.

[17] K. Yu, G. Guan, and M. Zhou, "Resume information extraction with cascaded hybrid model," in *ACL'05*, 2005, pp. 499–506.

[18] M. Merler, L. Cao, and J. R. Smith, "You are what you tweet pic! gender prediction based on semantic analysis of social media images," in *Multimedia and Expo (ICME), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[19] C. Lu, W. Lam, and Y. Zhang, "Twitter user modeling and tweets recommendation based on wikipedia concept graph," in *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[20] A. Farseev, L. Nie, M. Akbari, and T.-S. Chua, "Harvesting multiple sources for user profile learning: a big data study," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 2015, pp. 235–242.

[21] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "Cosnet: Connecting heterogeneous social networks with local and global consistency," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1485–1494.

[22] D. Downey, O. Etzioni, and S. Soderland, "A probabilistic model of redundancy in information extraction," DTIC Document, Tech. Rep., 2006.

[23] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web." in *IJCAI*, vol. 7, 2007, pp. 2670–2676.