Social Network Analysis and Mining (2018) 8:24
https://doi.org/10.1007/s13278-018-0495-0

ORIGINAL ARTICLE

Profiling Web users using big data

Xiaotao Gu[1] · Hong Yang[1] · Jie Tang[1] · Jing Zhang[2] · Fanjin Zhang[1] · Debing Liu[1] · Wendy Hall[3] · Xiao Fu[4]

Received: 16 January 2017 / Accepted: 16 February 2018 / Published online: 22 March 2018

Abstract
Profiling Web users is a fundamental issue for Web mining and social network analysis. Its basic tasks include *extracting* basic information, *mining* user preferences, and *inferring* user demographics (Tang et al. in ACM Trans Knowl Discov Data 5(1):2:1–2:44, 2010). Although methodologies for handling the three tasks are different, they all usually contain two stages: first identify relevant pages (data) of a user and then use machine learning models (e.g., SVM, CRFs, or DL) to extract/mine/infer profile attributes from each page. The methods were successful in the traditional Web, but are facing more and more challenges with the rapid evolution of the Web each persons information is distributed over the Web and is changing dynamically. As a result, available data for a user on the Web is redundant, and some sources may be out-of-date or incorrect. The traditional two-stage method suffers from data inconsistency and error propagation between the two stages. In this paper, we revisit the problem of Web user profiling in the big data era and propose a simple but very effective approach, referred to as MagicFG, for profiling Web users by leveraging the power of big data. To avoid error propagation, the approach processes all the extracting/mining/inferring subtasks in one unified framework. To improve the profiling performance, we present the concept of contextual credibility. The proposed framework also supports the incorporation of human knowledge. It defines human knowledge as Markov logics statements and formalizes them into a factor graph model. The MagicFG method has been deployed in an online system AMiner.org for profiling millions of researchers—e.g., extracting E-mail, inferring Gender, and mining research interests. Our empirical study in the real system shows that the proposed method offers significantly improved (+ 4–6%; $p \ll 0.01$, t test) profiling performance in comparison with several baseline methods using rules, classification, and sequential labeling.

Keywords Information extraction · Factor graph model · User profiling · Big data

1 Introduction

Web user profiling involves building a semantics-based user profile (consisting of contact information, educational history, demographics, and preferences/interests, etc.) from the

Xiaotao Gu and Hong Yang have contributed equally to this work.

✉ Jie Tang
jietang@tsinghua.edu.cn

Xiaotao Gu
gxt13@mails.tsinghua.edu.cn

Hong Yang
yangh13@mails.tsinghua.edu.cn

Jing Zhang
zhang-jing@ruc.edu.cn

Fanjin Zhang
zfjsail@smail.nju.edu.cn

Debing Liu
debingliu@tsinghua.edu.cn

Wendy Hall
wh@ecs.soton.ac.uk

Xiao Fu
xiao.fu@huawei.com

[1] Department of Computer Science, Tsinghua University, Beijing, China

[2] Department of Computer Science, Renmin University, Beijing, China

[3] Huawei Technologies Co. Ltd., Shenzhen, China

[4] Electronics and Computer Science, University of Southampton, Southampton, UK

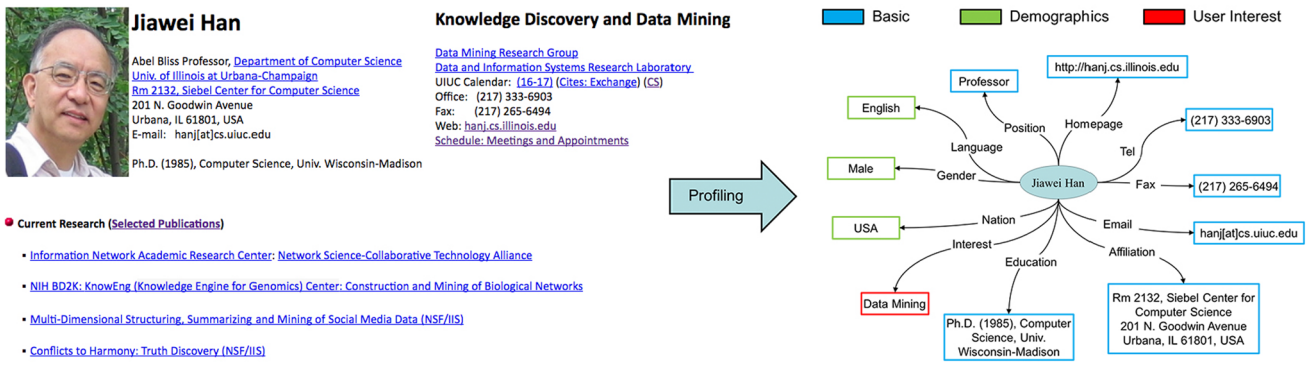


Fig. 1 Example of researcher profile. The profile contains extracted basic information such as affiliation, Position, picture, E-mail, etc., in blue rectangles; mined research interests in red rectangles; and inferred demographics in green rectangles (color figure online)

unstructured Web. It is a fundamental issue for understanding user behavior on the Web, and has long been viewed as an important and challenging problem in Web mining and social network analysis. The constructed user profiles can be applied to many applications, and the extraction process has become a necessary part of most online systems. For example, in a recommender system, with a large and high-quality profile database, we can make accurate recommendations to a user on what kind of information she/he would be interested in. In e-commerce, user profiles are extremely important for locating customers for a new product.

The basic tasks for profiling Web users include extracting basic information, mining user preferences, and inferring user demographics (Tang et al. 2010). As the Web may have various unstructured pages to introduce a user, the goal of extracting basic information is to extract structured profile attributes from the unstructured pages (Banko et al. 2007; Tang et al. 2007; Weninger et al. 2010; Wu et al. 2015). To understand user behavior, one common situation is that a system has collected a lot of user behavioral data for mining users' preferences/interests (Brajnik et al. 1987; Ikeda et al. 2013; Li et al. 2014; Pazzani and Billsus 1997; Soltysiak and Crabtree 1998; Wu et al. 2016). Demographics play an important role in revealing the reasons behind users' behavior and recently have attracted a lot of attention from both academic and industrial communities (Sarraute et al. 2015; Dong et al. 2014; Krulwich 1997). The three tasks were usually studied and many different methods have been proposed to deal with each of them, separately.

Despite slight differences, the general process of the different methods consists of two stages: first, identify relevant user pages (information), and then use machine learning models (e.g., SVM, CRFs, or DL) to extract/mine/infer profile attributes from the page. However, such two-stage methods suffer from data inconsistency and error propagation between the two stages. The problem of data inconsistency has become more and more critical due to the rapid evolution of the Web. As a result, a given persons information may be distributed on the Web and changing dynamically. The error propagation problem is due to the two stages of the methods; an error in the first stage will be propagated to the second stage. Figure 1 gives an example of building a researcher ("Jiawei Han") profile from the Web. A usual approach is to first find relevant Web pages such as homepage of Dr. Jiawei Han's homepage 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 is around 90.0%, respectively. For example, an F1 score of 92% is reported for the task of homepage finding conducted by Tang et al. (2008), and 87% for extracting the profile attributes from the homepage (Tang et al. 2010). However, taking error propagation into consideration, the overall accuracy of the approach that combines the two stages drops to 80%. More seriously, with the rapid growth of the Web, the problem becomes even more critical, as the big Web contains much more noisy data.

In this paper, we conduct a systematic study to revisit this problem from the perspective of big data. The essential question we want to answer is whether and how we can avoid error propagation by leveraging the power of big data. This is very challenging and raises a set of unique challenges.

- First, to avoid error propagation, we need to combine the two stages in traditional methods into one step. It is unclear how to develop a unified approach framework for processing all the profiling tasks together in one step.
- Second, though the data volume in the big Web is huge, it is unclear how to leverage its power wisely. The approach of first collecting all Web data and then processing it offline is obviously infeasible.
- Last, how to incorporate human knowledge into the profiling procedure. Numerous studies (Finkel et al. 2005; Tang et al. 2013) have shown that human knowledge can play a very important role in information extraction and

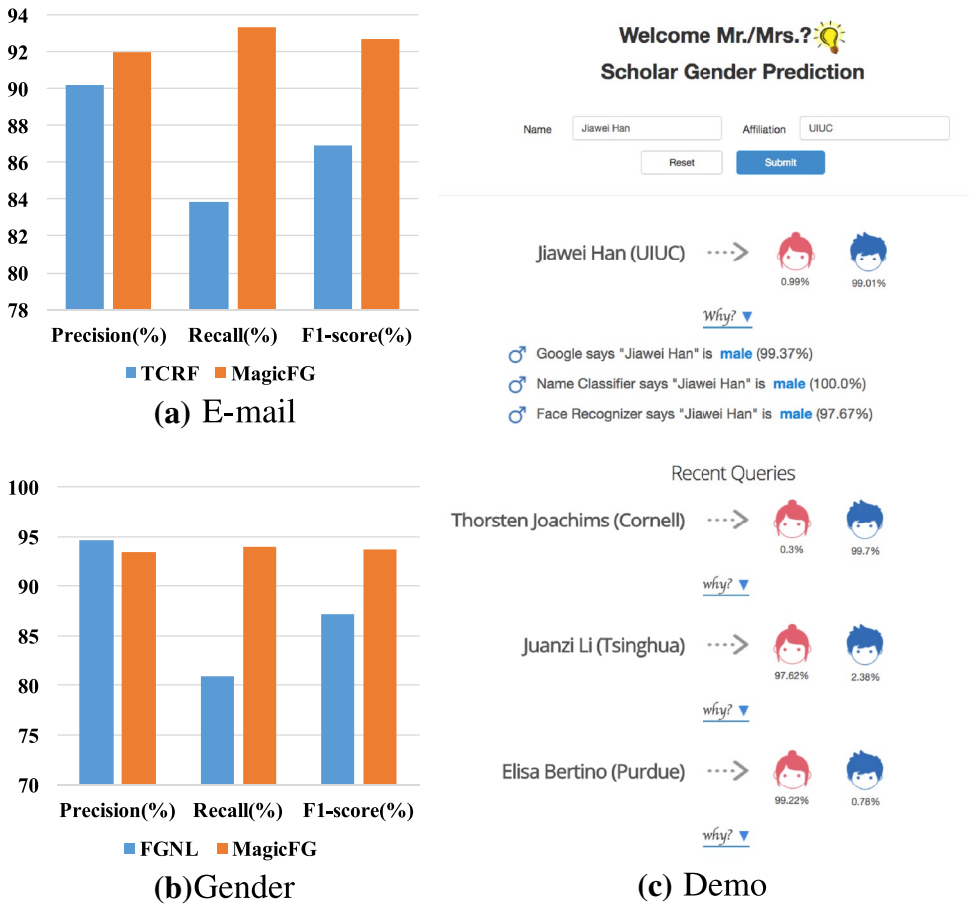


Fig. 2 Performance comparison between our approach and existing methods. **a** Comparison with TCRF (Tang et al. 2010), a two-step method for E-mail extraction. **b** Comparison with FGNL (Tang et al. 2011) for Gender inference (Cf. Sect. 4 for detailed comparisons). **c** A screenshot of the the scholar Gender prediction system

also other data mining tasks. It would be very helpful if the profiling approach were flexible enough to incorporate human prior knowledge.

The technical contribution of this work lies in the proposal of a novel approach framework, referred to as MagicFG, for profiling Web users. MagicFG offers a simple but very effective way to process all the profiling subtasks together, by leveraging the power of big Web data. One key idea here is that we do not collect original data from the Web. Instead we use a search engine to retrieve important snippets. For each retrieved snippet, we design contextual features to model its credibility. To further incorporate human knowledge, MagicFG provides a mechanism of formalizing human knowledge as Markov logics into a factor graph.

The MagicFG method has already been deployed in an online system AMiner.org,[1] for profiling millions of researchers—e.g., extracting E-mail, inferring Gender, and mining research interests. Our empirical study in the real system shows that the proposed method significantly improves (+ 4–6%; $p \ll 0.01$, t test) the profiling performance in comparison with several state-of-the-art methods using rules, classification, and sequential labeling (Cf. Sect. 4 for detailed comparisons). Figure 2 shows an example. We can see that our approach significantly outperforms the comparison methods for the tasks of E-mail extraction and Gender inference.

Besides directly integrating the profiling function into the AMiner system, we also developed and deployed a separate Web application for scholar Gender prediction based on the proposed MagicFG model. Figure 2c shows a screenshot of the scholar Gender prediction system. The system trains a MagicFG model off-line using existing labeled data in our dataset. When a user inputs a scholar name with her/his affiliation, the system calls three analyzers face analyzer, Google analyzer, and name analyzer and finally make prediction based on their weighted votes.

Organization In Sect. 2, we review the related work. In Sect. 3, we describe the proposed approach to user profiling. In Sect. 4, we present experimental results to evaluate

[1] https://aminer.org, AMiner aims to understand scientific text and networks. The system extracts researchers profiles automatically from the Web. So far, the system has built more than 130,000,000 researcher profiles and provides a set of unique functions, including expert search, social influence analysis, collaboration recommendation, and community evolution. The system has been in operation since 2006 and has attracted more than 8,320,000 independent IP accesses from over 220 countries/regions.

the effectiveness of the proposed approach. Finally, Sect. 5 concludes the paper.

2 Related work

As a key component in Web mining and social network analysis, the problem of profiling Web users has attracted considerable attention from both industry and academia. For example, E-mail Breaker,[2] E-mail Hunter,[3] and Sidekick[4] are three online services offering services to find E-mail addresses of requested people from the Web. Gender API[5] and Genderize[6] are two online APIs offering services to determine the Gender of a name. In this section, we review related scientific literature on the three profiling tasks: *profile extraction*, *demographics inference*, and *user preference mining*.

2.1 Profile extraction

Lots of effort have been invested in extracting user profile attributes from the Web. One particular type of related research is to extract profiles from a given set of documents. For example, Yu et al. propose a cascaded information extraction framework for identifying personal information from resumes (Yu et al. 2005). The basic idea is to first segment an input document into consecutive blocks and then apply a machine learning model to extract detailed profile information such as Address and E-mail. Artequakt (Alani et al. 2003) is a system using GATE (Cunningham et al. 2002), a rule-based extraction tool, to extract entity and relation information from the Web. Kristjansson et al. developed an interactive information extraction system to assist the user to populate a contact database from E-mails (Kristjansson et al. 2004). See also Balog et al. (2006). However, this thread of research is limited by the input documents and the proposed method is more or less ad hoc.

Recently, more and more attention has been paid to extracting profiles from the generic Web. For instance, Michelson and Knoblock (2007) propose a unsupervised method to extract information from the Web. Tang et al. (2010) employ a classification model to identify relevant pages for Web users and then use Tree-structured Conditional Random Fields (TCRF) (Tang et al. 2006) to tag tokens in profile pages and extract profile information. Weninger and Han (2013) viewed the Web as a massive and decentralized database and propose a method called CETR to improve information extraction with the Web. Several other similar studies can be also found in Banko et al. (2007) and Ritze et al. (2016). Despite slight differences, the general process of the different methods consists of two stages: first identify relevant pages (information) of a user and then use machine learning models (e.g., SVM, CRFs, or DL) to extract/mine/infer profile attributes from the page. However, most of aforementioned methods perform the extraction in two stages, and thus suffer from error propagation between the two stages.

Regarding the extraction models, many have been proposed in the past 30 years. Hidden Markov Model (HMM) (Ghahramani and Jordan 1997), Maximum Entropy Markov Model (MEMM) (McCallum et al. 2000), Conditional Random Field (CRF) (Lafferty et al. 2001), Partially Labeled Factor Graph Model (Tang et al. 2011), Transfer Factor Graph Model (Tang et al. 2016), Support Vector Machines (SVM) (Cortes and Vapnik 1995), and Voted Perceptron (Collins 2002) are widely used models. Sarawagi and Cohen (2004) also propose a semi-Markov Conditional Random Fields for information extraction. An overview of the existing information extraction models is given in Tang et al. (2007).

2.2 Demographics inference

User demographics offer the potential to better understand user behavior and the interactions between users. However, in practice, the available demographic information in many online systems is very limited. There are several works attempting to inferring user demographics automatically, for example, based on their online browsing (Hu et al. 2007), gaming (Szell and Thurner 2012) and search (Bi et al. 2013) behaviors. In social network analysis, several efforts have been also made to infer Age (Sarraute et al. 2015), location (Li et al. 2012; Efstathiades et al. 2016), and identity (Joseph et al. 2016). Dong et al. (2014) propose employing a factor graph model to simultaneously predict Age and Gender based on user communication patterns in mobile networks. However, all these works study the demographics inference problem on a specific dataset.

A few other studies aiming to improve the inference accuracy of demographics using multiple sources. For example, Li et al. (2012, 2014), and Ikeda et al. (2013) use Facebook, Google Plus, and Twitter to improve the accuracy of demographics inference. Compared to these works, the problem studied in this work is more open—we attempt to infer user demographics using only person name and the potential information from the Web. In terms of this, the most similar work is Tang et al. (2011), which presents a name-centric approach to make Gender inference. Their idea is to create a comprehensive and contemporary name list from Facebook,

[2] http://emailbreaker.com.
[3] https://emailhunter.com.
[4] http://www.getsidekick.com.
[5] https://www.gender-api.com/.
[6] https://genderize.io/.

and then use the name list to help improve Gender inference performance. Another similar research can be also found in Eltaher and Lee (2015). The proposed approach in this work does not heavily reply on the quality of a name list. It combines all the available information from the big Web and achieves a much better inference performance.

2.3 User preference mining

The other type of research is to learn user preference from user behavioral data (e.g., behavioral logs) or user-associated documents. For example, Pazzani and Billsus (1997) propose algorithms for learning user profiles and use the profile to determine whether a user would be interested in World Wide Web sites on a specific topic. Chan (1999) has developed a personalized Web browser. It learns a user profile, and aims at helping users navigating the Web by searching for potentially interesting pages for recommendations. Soltysiak and Crabtree (1998) describes an experimental work to study whether user interests can be automatically classified through heuristics. More recently, Makazhanov et al. (2014) propose a method to predict user interests from Twitter data. Figueiredo et al. (2016) aim to mine user preferences from the trajectory data. Wu et al. (2016) jointly model users' temporal behaviors that indicate their preferences and social links in a probabilistic approach. Our proposed approach can be applied to not only user preference mining, but also the other tasks (profile extraction and demographics inference) for Web user profiling.

2.4 Data redundancy

Data redundancy is another relevant research area relevant to this work. The correlation between data redundancy and the probability of correctness in information extraction has been studied in Downey et al. (2005). Later, Pedro et al. (2011) analyzed the dependencies between overlapping or duplicated content in videos and propose a novel tag propagation method for automatically obtaining richer video annotations. Blanco et al. (2010) exploit the data redundancy to automatically extract and integrate data from the Web and validate the feasibility and effectiveness of this approach. In general, most existing works on data redundancy try to remove the redundant information. In this work, we attempt to utilize data redundancy to support Web user profiling.

3 Approach framework

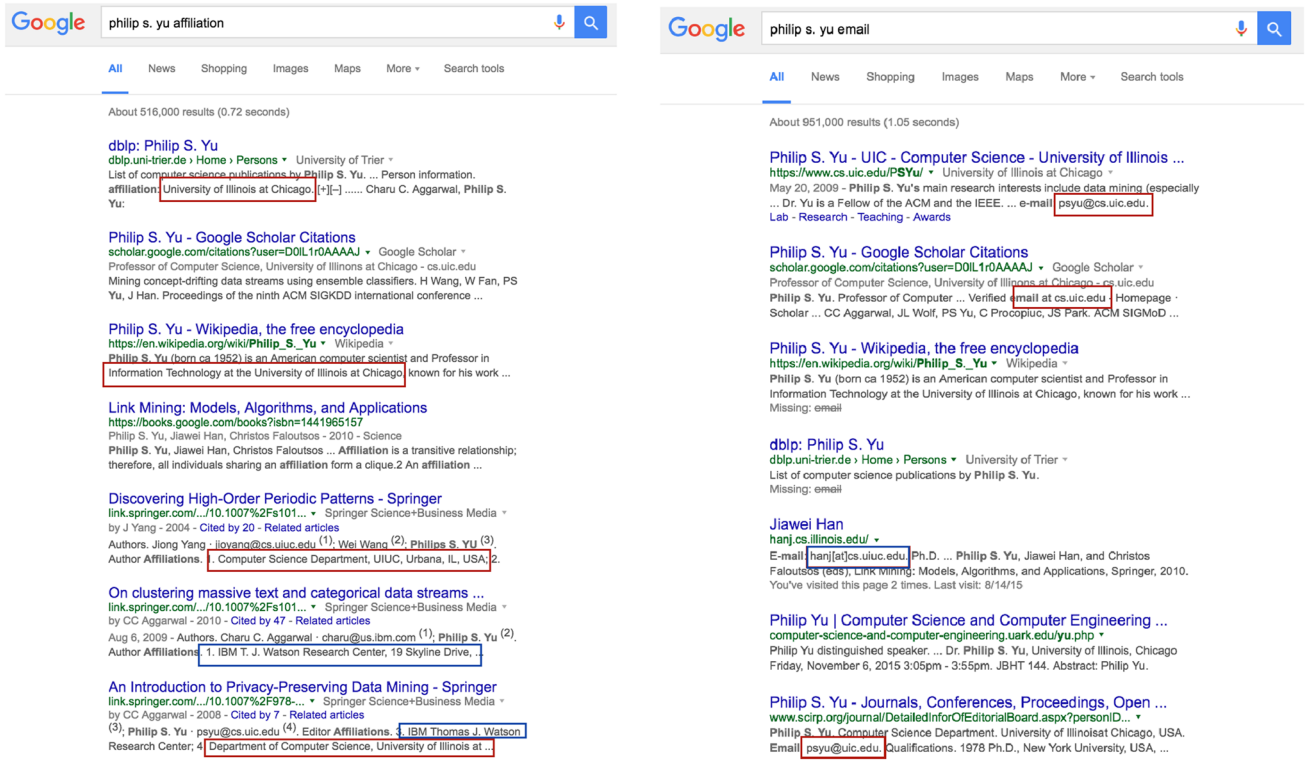
In this section, we first give the basic idea of the proposed framework to solve the profiling problem. In particular, we focus on how to profile researchers on the Web. We introduce three methods to extract profile attributes from the Web.

3.1 Basic idea

Given a person v, referred to as a query person, our goal is to extract profile attributes of the person and construct a researcher profile. For example, in an academic search system, e.g., AMiner.org (Tang et al. 2008), the researcher profile consists of Position, picture, address, phone, E-mail, homepage, research interest, etc. A detailed definition can be found in Tang et al. (2010). Our goal is to design a general method to automatically extract the profile attributes from the Web with high accuracy. Meanwhile, the method should be also flexible enough to be 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% (Tang et al. 2010; Tang et al. 2008). However, the overall accuracy achieved when combining the two steps inevitably drops to 80%, due to error propagation between the steps. Meanwhile, the required profile attributes may be distributed over different Web pages, which results in two new problems how to perform extraction from distributed pages and how to perform 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. Then an extraction model is applied to the returned snippets to extract the profile attributes. The idea behind is to leverage data redundancy to support the extraction. Suppose we are going to extract the affiliation of "Philip S. Yu." The constructed query can be "Philip S. Yu affiliation." Similarly, to extract the E-mail address of "Philip S. Yu," we can construct "Philip S. Yu E-mail." Figure 3 shows two example snippets returned by Google with two constructed queries. We see that we can easily identify two different affiliations: "University of Illinois at Chicago" and "IBM T. J. Watson Research Center" (after normalization) from Fig. 3a, and two E-mail addresses: "psyu@cs.uic.edu" and "hanj[at]cs.uiuc.edu" from Fig. 3b. We call the identified affiliations/E-mails from the snippets as *candidate* affiliations/E-mails. Now the problem becomes how to recognize which candidate is correct and which one is not. We formalize this problem as a ranking problem. Specifically, our idea here is to leverage the redundancy information—e.g., "University of Illinois



(a) Affiliation **(b)** E-mail

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 E-mail addresses

at Chicago" occurs four times in the snippets and "IBM T. J. Watson Research Center" occurs twice. More precisely, we propose a MArkov 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 improve the extraction accuracy.

It is noteworthy that we are not restricted to the two example attributes. The proposed method itself 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 model. For some profile attributes, it is easy to construct the query. For example, we found that for E-mail, we can achieve a high accuracy by simply using name + E-mail. For some other profile attributes—for example, Gender and Position—the situation may be more complicated. We will introduce how we construct a smart query for general profile attributes in Sect. 3.2. Please also note that there are generally two types of profile attributes: *categorical* attributes and *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 E-mail and Age are non-categorical attributes.

The two different types of attributes will be treated slightly differently in the proposed framework.

3.2 Smart query construction

For categorical attributes, we construct the query by automatically identifying representative keywords in each candidate category and combining them together to form a query. To find the representative keywords for each category, we first collect a number of 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 (Baeza-Yates and Ribeiro-Neto 1999). The TF-IDF score of a word w in a category c is calculated as follows:

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

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 E-mail extraction can be "name E-mail|mail."

3.3 Baseline models

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

3.3.1 Rule-based model

In the rule-based model, for extracting the E-mail of the query person v, we simply construct the query by combining the person name and the word "E-mail." After obtaining the returned snippets from a search engine (e.g., Google), we can use rule-based heuristics to extract the E-mail of the query person. Specifically, we first extract the candidate E-mail addresses from the searched snippets; if the first name or the last name of the query person is contained in the prefix of a candidate E-mail address, then the extracted E-mail will be selected as the result. To recognize the E-mail addresses, we define a regex pattern.[7] This is because in many Web pages, especially on some person's homepages, the E-mail addresses may be encoded in different ways such as "firstname [dot] lastname [at] cmu [dot] edu."

Our preliminary experiments show that such a simple method could result in an accuracy of 88%—comparable with the state-of-the-art performance obtained by a traditional two-step approach (Cf. Sect. 4 for detailed comparisons).

Limitations The rule-based method usually results in a high precision, but low recall. Moreover it cannot distinguish the importances of different patterns (rules).

3.3.2 Classification-based model

We can train a classification model to learn the weights of different rules/patterns, if we define more than one rule. Let us first consider a two class classification problem. Let $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ be a training data set, in which x_i denotes a feature vector of a candidate information packet 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 what 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. Sect. 3.4 for details). In our experiments, we use logistic regression (LR) as the classification model. The classification can adjust the weights of different features (patterns) and combine the feature together, thus obtains a better performance (90% in terms of F1-score) than the rule-based method.

Limitations The classification-based method still does not consider the correlations between different candidates. As shown in Fig. 3, the returned snippets usually contain redundant information that might be helpful for the extraction. Both the rule-based and the classification-based models consider each candidate independently, and thus cannot leverage such redundant information.

3.4 Markov logic factor graph (MagicFG) model

In practice, the redundant information resided in the snippets can be very helpful for improving the extraction accuracy.

For example, in Fig. 3b, for E-mail extraction, the same prefix "psyu" before "@" in the two candidate E-mail addresses, "psyu@cs.uic.edu" and "psyu@uic.edu," indicates that the two E-mail addresses belong to the same person.

To model and incorporate the redundancy-based correlation into a unified profiling model, we propose a Markov logic factor graph (MagicFG) model by formalizing the correlations as first-order logic statements. We now introduce how to model the data redundancy for the non-categorical and categorical attributes, respectively.

Modeling non-categorical attributes For each query person, we construct a factor graph model with each node representing a candidate instance, and each edge corresponding to a dependency between two candidates. We optimize the factor graph model for all query persons simultaneously. We explain the modeling process of categorical attributes using E-mail as an example. For query person v, we denote each candidate E-mail as e_i. As the example in Fig. 4, we could extract four candidates $\{e_1, e_2, e_3, e_4\}$. For each candidate E-mail, we create an instance (e_i, v) and associate it with a

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

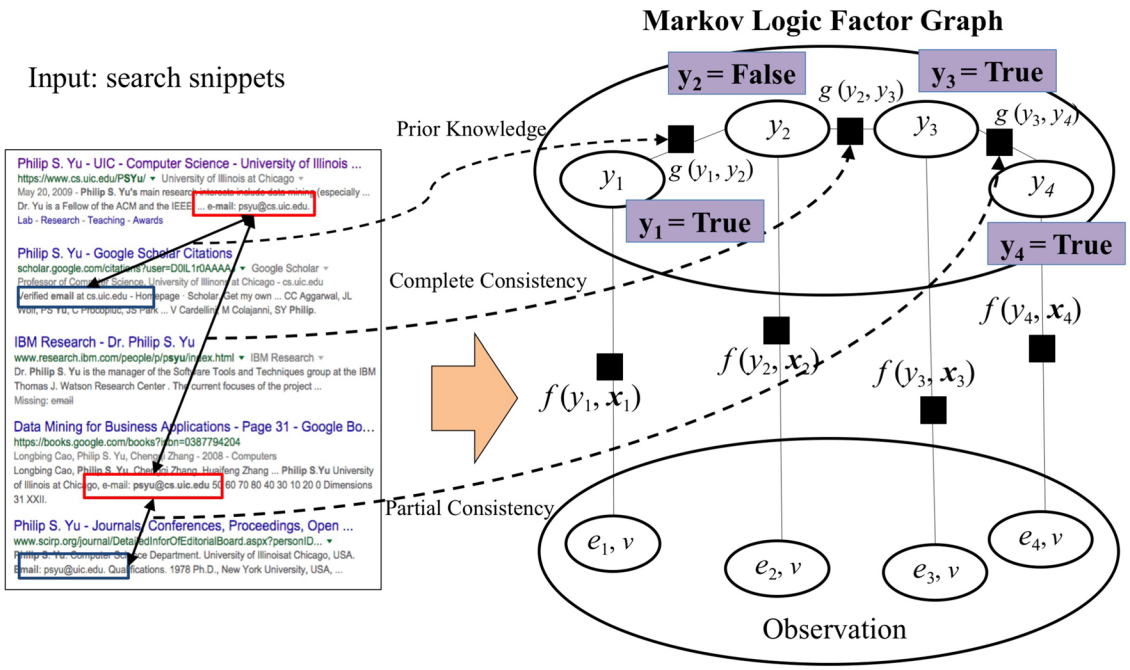


Fig. 4 Graphical representation of a logic factor graph model based on a real search example. Notation (e_i, v) represents an E-mail–person pair, and y_i indicates its corresponding label; notations $f(.)$ and $g(.)$ represent the attribute factor function and logic factor function, respectively

latent variable y_i. To model the correlations between candidate instances, we construct a factor graph model as illustrated in Fig. 4. The model is referred to as a Markov logic factor graph (MagicFG) model. In MagicFG, each correlation is represented as a first-order logic statement. The correlation can be prior knowledge or any other human-defined correlations. We will explain how we define the first-logic based correlation later. At the higher level, in MagicFG, we define two types of factor functions.

- *Attribute factor function* Captures characteristics of the E-mail–person pair and is defined as an exponential function:

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

  where $\phi_k(.)$ is the kth 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 ith feature vector. Z_a is the normalization factor.
- *Logic factor function* Captures the correlations between latent variables. It is also defined as an exponential function:

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

  where $\psi_m(.)$ is the mth 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 E-mail, we define features such as whether v's first name, last name or full name is contained in e_i's prefix.[8] 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 human domain human knowledge about the extraction. For a general introduction to first-order logic, please refer to Richardson and Domingos (2006).

[8] We call the string before "@" of an E-mail candidate as the prefix of the E-mail, and the string after "@" as its domain.

Table 1 First-order logic knowledge base

First-order logic	Example
Complete consistency	$\text{Equals}(e_i, e_j) \Rightarrow \text{Equals}(y_i, y_j)$
Partial consistency	$\text{SamePrefix}(e_i, e_j) \Rightarrow \text{True}(y_i) \wedge \text{True}(y_j)$
Prior knowledge	$\text{IsBlocked}(e_j) \wedge \text{SameDomain}(e_i, e_j) \Rightarrow \text{True}(y_i) \wedge \text{False}(y_j)$

Complete consistency describes the requirements 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 statement

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

indicates that y_i equals y_j if the corresponding E-mail candidates are the same with each other. The logic is straightforward because two identical E-mail addresses are highly likely to be credible or not at the same time. Correspondingly, we define the factor function as

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

Partial consistency describes a situation in which the values of two latent variables y_i and y_j are partially consistent under some given conditions. For example, the following first-order logic statement

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

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

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

Prior knowledge referes to knowledge that can be formalized into useful first-order logics for a specific task. For example, when we search for someone's E-mail address using Google, we find that many candidates starting with "E-mail" like "email@gmail.com." This is due to the security policy of the search engine, which hide the actual E-mail address (called a blocked candidate). Fortunately, we can still observe the domain information.

We found that when another candidate shares the same domain with a blocked candidate, it is very likely that the other candidate is a correct E-mail—we use the redundant domain information to enhance the confidence. 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

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

For each profiling task, we build a knowledge base according to the defined first-order logics and summarize it in Table 1. In general, the attribute factors capture the characteristics on each potential person–E-mail pair and the logic correlation factors capture the dependencies between two person–E-mail 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 each edge representing the dependency 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 combining 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 share the same first name.

In summary, the factor graphs built for the non-categorical and categorical attributes are slightly different. For non-categorical attributes, we build multiple graphs, where each graph corresponds to a person with a node in the graph representing a candidate attribute and an edge representing the dependency between two connected candidate attributes. While for categorical attributes, we build only one graph, where a node represents a person and an edge represents the dependency between two persons.

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 (Hammersley and Clifford 1971):

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

where Z is the normalization factor; $e_i \sim e_j$ indicates that there is a (direct or indirect) 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 involves estimating 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:

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

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) (Yedidiaet al. 2000) 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 :

$$\theta_{\text{new}} = \theta_{\text{old}} + \eta \cdot \frac{\partial L(\theta)}{\partial \theta}, \tag{7}$$

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 $Y_v = \{y_1, \ldots, y_I\}$ for all the person–E-mail pairs $\{(e_i, v)\}$:

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

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 is a challenging task itself. Moreover, the profile information is very dynamic. Our method avoids this problem by directly querying the search engine.

3.5 Enhancement for Gender inference

The proposed MagicFG framework is very flexible and can be easily extended (enhanced) for particular tasks. We use Gender inference as an example to explain how we enhance the MagicFG model by various additional information. We have developed a Gender prediction system by enhancing MagicFG with several new features.

Face recognizer (FR) Besides searching only Google snippets, we also use Google Images to retrieve redundant image information. Then we apply a face recognition tool to recognize if there is a male/female face in the image(s).[9]

Facebook generated name list (FGNL) We also consider the Facebook Gender list. The list was collected by Tang et al. (2011). The method is also one of comparison methods in our experiments. To enhance our method for Gender inference, we define FGNL as a factor function in our MagicFG model.

Super name list (SNL) This is a refined version of FGNL. From all listed names in FGNL, we first extract all n-gram subwords $\{s_{ni}|i = 1, 2, \ldots, m\}$. We denote S_n^c as the pool of subwords appearing in the name list for Gender $c \in \{\text{male}, \text{female}\}$, and $\text{count}(s_{ni}^c)$ as the number of appearances of s_{ni} in S_n^c. The subword density $\rho(s_{ni}, c)$ is calculated as

$$\rho(s_{ni}, c) = \frac{\text{count}(s_{ni}^c)}{\sum_{j=1}^{m} \text{count}(s_{nj}^c)}. \tag{9}$$

For the target user v with full name f_v, the prediction is made by

$$g(v) = \begin{cases} \text{male}, & \text{if score(male)} > \text{score(female)} \\ \text{female}, & \text{otherwise} \end{cases} \tag{10}$$

where

$$\text{score}(c) = \sum_{i=1, s_{ni} \in f_v}^{|S_n^c|} \rho(s_{ni}, c), \; c \in \{\text{male}, \text{female}\} \tag{11}$$

In summary, SNL tries to capture local patterns of a name that may contain Gender information.

MagicFG++ We combine all the defined factor functions together. To distinguish from the general MagicFG model, we call this enhanced method for Gender inference as MagicFG++.

In our experiments, we will evaluate the enhancement method and compare the different methods.

4 Experiments and discussions

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 E-mail 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.[10]

4.1 Experiment setup

Dataset To construct a ground-truth dataset for quantitative evaluation, we randomly choose 2000 researchers from AMiner.org (Tang et al. 2008). Specifically, for extracting the E-mail of each researcher, we search the Web using a search engine by querying the person name and the word "E-mail." This way results in 4528 E-mail candidates. Human annotations are applied to identify correct E-mail addresses. Analogously, for inferring Gender, we search the Web by querying the person name and the word "his" or "her." Human annotations are also used to identify the Gender of 2700 researchers. For disagreements in the annotation, we conduct "majority voting." We found that, among the 2700 researchers, 47.5% are female researchers;[11] and about 40% of the E-mail candidates are correct E-mails.

Evaluation metrics To quantitatively evaluate the proposed model, we perform fivefold cross-validation and report the average extraction performance in terms of precision, recall, and F1-score.

Comparison methods We compare our MagicFG method with following methods for extracting E-mail and Gender on the ground-truth dataset.

- *Rule* We use several pre-defined rules to extract profile attributes. For example, for extracting Gender, we count the number of common names for girls and boys. For extracting E-mails, we find whether the prefix of the E-mail contains the person name.
- *Random forest (RF)* We use the same attribute factors as features and employs sklearn package to conduct train and predict.
- *Logistic regression (LR)* We use the same attribute factors as features and employs sklearn package to conduct train and predict.
- *Support vector machine (SVM)* We use the same attribute factors as features and employs the sklearn package to conduct train and predict.

[9] In our experiments, we use Face++, http://www.faceplusplus.com/.

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

[11] The Gender dataset is larger than the previously released one with more balanced distribution of two Genders.

Table 2 Feature definition of E-mail extraction

Feature type	Description
Domain-specific	(*2) Whether the E-mail username contains v's last/first name
Contextual	(*1) The ranking Position of the source page snippet
	(*2) Whether the source page snippet/title contains v's affiliation
	(*4) Whether the source page snippet/title contains v's last/first name

Table 3 Feature definition of Gender inference

Feature type	Description
Domain-specific	(*2) Term frequency of "his/her" among all snippets
	(*2) Document frequency of "his/her" among all snippets
Contextual	(*2) Whether "his/her" appeared in top k snippets
	(*4) Co-occurrences of "his/her" and v's affiliation in the source page snippet/title
	(*8) Co-occurrences of "his/her" and v's last/first name in the source page snippet/title

- *Tree-structured conditional random field* For E-mail extraction, we use the method proposed in Tang et al. (2010) as the baseline (to hereafter referred to as: TCRF), which is one of the state-of-the-art approaches to extracting homepages and E-mails from the Web. This method has two steps, where it first finds the user's homepage and then extracts E-mail from the homepage with a high precision using the TCRF model.
- *Facebook generated name list predictor (FGNL)* For Gender inference, we use a method proposed by Tang et al. (2011) as the baseline (to hereafter referred to as: FGNL). Most state-of-the-art methods for inferring Gender depend on a list of common names for males and females. In Tang et al. (2011), the authors proposed an approach that 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 a male name, the user is treated as a male, and similarly for females. 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.

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, and conduct a fivefold cross validation for each method.

Table 4 Performance comparison of E-mail extraction (%)

Method	Precision	Recall	F1-score	AUC
TCRF	90.20	83.83	86.90	–
Rule	87.81	89.64	88.72	–
RF	90.05	89.42	89.74	93.09
LR	91.97	89.83	90.89	94.42
SVM	90.58	90.21	90.40	93.14
MagicFG	94.27	92.90	93.58	97.38

4.2 Feature definition

We now turn to the definition of attribute factor functions. For the problem of Web user profiling, there are mainly two types of features. The first are domain-specific features, which differ in different extraction subtasks. For example, in E-mail extraction, we could define binary-valued features to describe whether the query person's first or last name is contained in the E-mail address. In Gender inference, we could use term frequencies of keywords "his/her" among search snippets as real-valued features.

The second type of features are contextual features to model contextual credibility of search snippets from which we extract all domain-specific features. These features try to model whether these snippets are relevant to the query person. Contextual features mainly consist of the ranking Position of the search snippet as an integer or as the ratio of its ranking to the total number of shown snippets in one search page, and whether the snippet contains key information about the query person, such as her name or affiliation terms.

Tables 2 and 3 give more details on how we define these features, where "*k" indicates there are k features from different combinations of alternative factors in the description.

4.3 Profiling performance

E-mail extraction Table 4 lists the performance comparison of E-mail extraction. Our methods consistently ourperform the baseline. Among all methods, MagicFG with full features (Magic-C) significantly stands out in AUC and F1-score (achieves an improvement of + 6.68% compared with TCRF, $p \ll 0.01$ with t test). It is noteworthy that, aside from achieving better precision, our method shows clearly

Table 5 Performance comparison of Gender inference (%)

Method	Male				Female			
	Precision	Recall	F1-score	AUC	Precision	Recall	F1-score	AUC
FGNL	92.28	82.53	87.12	–	85.66	93.80	89.54	–
Rule	86.49	87.14	86.81	–	88.30	87.41	87.85	–
RF	91.03	88.48	89.74	96.17	89.82	92.10	90.95	96.17
LR	92.20	91.98	92.09	97.41	92.75	92.95	92.85	97.41
SVM	91.90	92.68	92.29	97.54	93.32	92.60	92.96	97.54
MagicFG	95.56	95.34	95.45	98.91	95.80	96.00	95.90	98.91

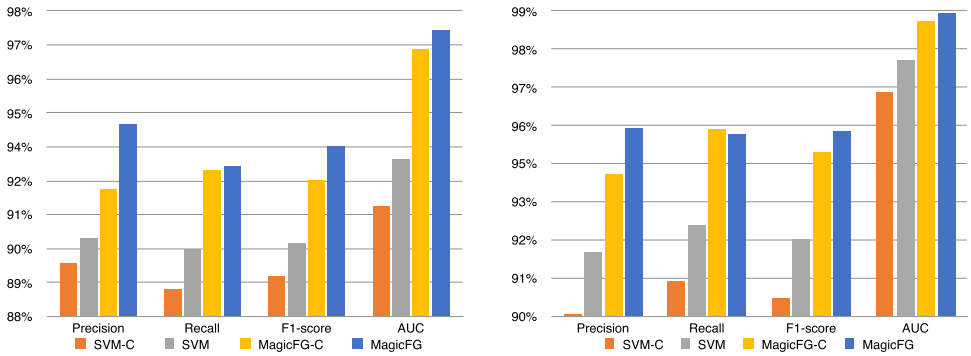


Fig. 5 Effect of contextual features in E-mail extraction and Gender inference. Methods ending with ″-C″ ignores contextual features, and uses domain-specific features only

(a) E-mail **(b)** Gender

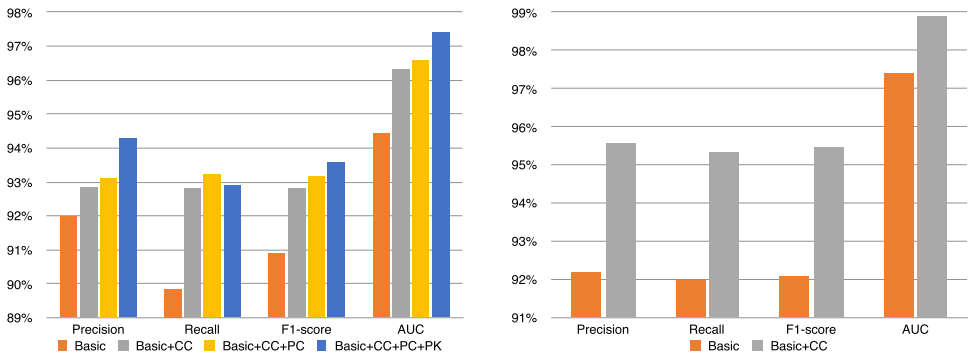


Fig. 6 Effect of logic correlation factors in E-mail extraction and Gender inference. Basic stands for the MagicFG model that only considers the attribute factors. +CC stands for adding the factors of complete consistency logic. +PC adds factors of partial consistency logic. +PK adds the factors of prior knowledge logic

(a) E-mail **(b)** Gender

better recall performance (+ 9.07%) than the baseline. This is due to its ability to find more relevant candidates through efficient query construction (Cf. Sect. 3.2), and to leverage the correlations between candidates.

Gender inference Table 5 lists the performance comparison of Gender inference. The proposed MagicFG outperforms the baseline (FGNL) in terms of F1-score by + 8.33%. The significant improvement comes mainly from the improvement on recall (+ 12.81%). The reason is that the FGNL method depends heavily on the quality of the name list and is thus limited by its coverage. On the other hand, our approach automatically identifies representative keywords for documents describing a user with specific Gender, and infer Gender from the big Web data with better generalization.

Effect of contextual features Figure 5 shows performances of SVM and MagicFG when they use or ignore contextual features. It is clear that contextual features are very useful in improving profiling performance, especially F1-score and AUC. This is done by reducing noises in Web data, such as basic information or demographical keywords extracted from irrelevant search snippets.

Table 6 Performance comparison of Gender inference (%)

Method	Male			Female		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Face	79.33	89.36	84.03	89.20	78.99	83.77
FGNL	92.28	82.53	87.12	85.66	93.80	89.54
SNL($n = 2$)	94.70	82.00	87.88	85.53	95.83	90.38
SNL($n = 3$)	94.38	86.52	90.26	88.75	95.34	91.92
SNL($n = 4$)	93.35	89.37	91.30	90.77	94.20	92.44
SVM	91.90	92.68	92.29	93.32	92.60	92.96
MagicFG	95.56	95.34	95.45	95.80	96.00	95.90
Genderize.io	96.22	88.38	92.13	90.05	96.76	93.26
MagicFG++	96.94	96.59	96.76	97.03	97.25	97.13

Effect of factors In both experiments of E-mail extraction and Gender inference, MagicFG stands out in performance compared with other classification methods. Here, we further present an in-depth analysis of how different logic correlation factors affect the performance of user profiling. Figure 6 shows the different evaluation metrics of the proposed MagicFG by considering different levels of logic factors. It can be clearly seen from Fig. 6a that for E-mail 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 6b also shows that the factor of complete consistency logic improves the performance of Gender inference significantly.

4.4 Discussion

Here we analyze the connection between MagicFG and several related models that can also be used for this extraction task for candidate verification.

Logistic regression Cox (1958) is widely used for classification problems. However, such models can hardly describe relationships between data points, which prove to be helpful in web information extraction tasks. In essence, MagicFG leverages logic factor functions to model such relationships among redundant data. Without the logic factor functions, MagicFG would be reduced to a naive logistic regression model.

HMRF-KMEANS Basu et al. (2004) aims at incorporating supervision into prototype-based clustering by defining ″must-link″ and ″cannot-link″ relationships between data points. Despite the similar idea of adding constraints to a probability framework, logical formulas are limited to describing exactly two types of relationships. Our proposed MagicFG model can handle a much wider range of logical relationships and can be easily generalized to higher-order logics.

Markov logic network (MLN) Richardson and Domingos (2006) is a classical model to combine a first-order logic knowledge base and probabilistic graphical models. It constructs a graph purely consisting of weighted formulas and can make inferences for unknown labels and relationships in an elegantly and powerfully way. However, logical formulas alone are insufficient for capturing complicated features. Consider Gender inference, for instance. We can design logic rules like: *In all relevant Google snippets, if "his" appears more frequently than "her," the user is labeled as a male one*, or in the form of a first-order logic formula:

$$\text{isGreater}(\#\text{his}, \#\text{her}) \rightarrow \text{isMale}(\text{user}).$$

Though seeming natural and intuitive, this rule is too simple to cover real scenarios. In fact, ″his″ is relatively more common in Web pages, and up to 16% of pages standing for female users showed more instances of ″his″ than ″her.″ MagicFG, however, is able to learn separate weights for these numerical features. In this case, MagicFG can find out how many ″her″s do we need to vote for ″female,″ while MLN cannot. MagicFG can be transformed into MLN when all attribute features are only described in first-order logic formulas.

4.5 Prototype system

We applied the proposed MagicFG model to the online system AMiner.org to extract researcher profiles. We have also developed several prototype systems. Here, we introduce one of them—Scholar Gender Prediction.[12]

Figure 2c shows a screenshot of the scholar Gender prediction system. The system trains a MagicFG model off-line using existing labeled data in our dataset. The user inputs a scholar name (“Jiawei Han”) and his affiliation (“UIUC”), and the system returns the prediction results (Male—99.01% and Female—0.99%). If the user is interested in more details,

[12] https://aminer.org/gender.

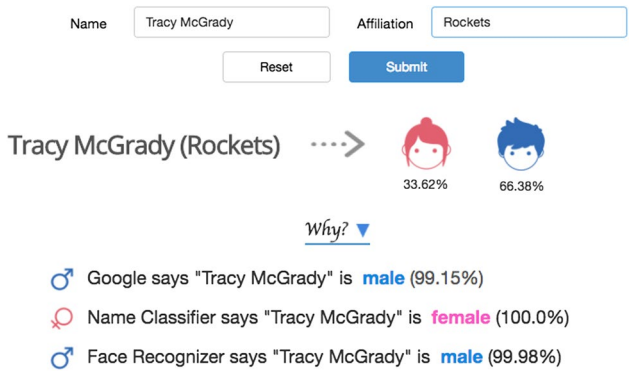


Fig. 7 Case study: Gender prediction for "Tracy McGrady from Rockets"

the system can display the results of different analyzers (factor functions) in MagicFG. We also provide related APIs, which offer programming-friendly interfaces for the AMiner system to access detailed profiles of scholars.

We now compare the system with two commercial systems: Gender API and Genderize.io. Both are popular Gender services. Genderize.io claims to have over 20 million distinct names across 79 countries and 89 languages. Table 6 lists the performance comparison between different systems using the same dataset used in our experiments. All the results are based on fivefold cross-validation. We also report results of the enhanced Gender inference (MagicFG++) and different Gender inference methods (Cf. Sect. 3.5). It seems that Genderize.io is better than FGNL, but still underperforms our proposed MagicFG model. By incorporating the enhanced factors, MagicFG++ can further improve the prediction accuracy.

In Fig. 7, we further give an example to demonstrate the generalization of proposed MagicFG model and to explain the unique advantages of MagicFG compared with the other systems (e.g., Genderize.io). In the example, we are trying to predict the Gender of "Tracy McGrady from Houston Rockets," who is a famous male basketball player. First the name is not in our AMiner database. When we try Genderize.io, the result is female, possibly because the first name or the last name is also common in females. Our model outputs a percentage of 33.62% to be female and 66.38% to be male. More importantly, our model can give a detailed explanation on the prediction result, as shown in Fig. 7. By both search engine and face recognition, the prediction results are male, while by only names, the prediction result is indeed female—this is probably why Genderize.io made an incorrect prediction.

5 Conclusion

In this paper, we revisited the problem of Web user profiling in the big data era and propose a simple but very effective approach, referred to as MagicFG, for profiling Web users by leveraging the redundant information on the Web. MagicFG also provides a mechanism to incorporate human knowledge as Markov logics into a factor graph. Experiments on two data sets show that the proposed method significantly improves the profiling accuracy in comparison with several comparison methods. The approach has been already deployed in an online system AMiner.org for profiling millions of researchers and mining research interests.

The proposed framework (MagicFG) has many potential applications. For example, we can apply the approach to help identify relationships between entities with the same idea. We can also extend the framework to extract entity attributes for building large-scale knowledge graphs. The general problem of profiling Web users represents an interesting research direction in Web mining and social network analysis. There are many potential future directions of this work. First, the information on the Web changes very quickly. How to recognize what is out-of-date and what is still valid is a challenging problem. Next, it is interesting to further study how incrementally learning the proposed model so that we can directly involve online user interactions in the learning process. Another potential direction is to study the profiling task using social data such as Facebook and Twitter data.

Acknowledgements The work is supported by the National Basic Research Program of China (2014CB340506), National Natural Science Foundation of China (61631013, 61561130160), a research fund supported by MSRA, and the Royal Society-Newton Advanced Fellowship Award.

References

Alani H, Kim S, Millard DE, Weal MJ, Hall W, Lewis PH, Shadbolt NR (2003) Automatic ontology-based knowledge extraction from web documents. IEEE Intell Syst 18(1):14–21

Baeza-Yates R, Ribeiro-Neto B (1999) Modern information retrieval. ACM Press, New York

Balog K, Azzopardi L, de Rijke M (2006) Formal models for expert finding in enterprise corpora. In: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, pp 43–55

Banko M, Cafarella MJ, Soderland S, Broadhead M, Etzioni O (2007) Open information extraction from the web. In: Proceedings of the 20th international joint conference on artificial intelligence, pp 2670–2676

Basu S, Bilenko M, Mooney RJ (2004) A probabilistic framework for semi-supervised clustering. In: Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining, pp 59–68

Bi B, Shokouhi M, Kosinski M, Graepel T (2013) Inferring the demographics of search users: social data meets search queries. In: Proceedings of the 22nd international conference on world wide web, pp 131–140

Blanco L, Bronzi M, Crescenzi V, Merialdo P, Papotti P (2010) Redundancy-driven web data extraction and integration. In: Proceedings of the 13th international workshop on the web and databases, pp 7:1–7:6

Brajnik G, Guida G, Tasso C (1987) User modeling in intelligent information retrieval. Inf Process Manag 23(4):305–320
Chan PK (1999) Constructing web user profiles: a non-invasive learning approach. In: KDD-99 workshop on web usage analysis and user profiling, pp 39–55
Collins M (2002) Ranking algorithms for named-entity extraction: boosting and the voted perceptron. In: Proceedings of the 40th annual meeting on association for computational linguistics, pp 489–496
Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
Cox DR (1958) The regression analysis of binary sequences. J Roy Stat Soc Ser B (Methodol) 20(2):215–242
Cunningham H, Maynard D, Bontcheva K, Tablan V (2002) GATE: a framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th annual meeting of the association for computational linguistics, pp 168–175
Dong Y, Yang Y, Tang J, Yang Y, Chawla NV (2014) Inferring user demographics and social strategies in mobile social networks. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, pp 15–24
Downey D, Etzioni O, Soderland S (2005) A probabilistic model of redundancy in information extraction. In: Proceedings of the 19th international joint conference on artificial intelligence, pp 1034–1041
Efstathiades H, Antoniades D, Pallis G, Dikaiakos MD (2016) Users key locations in online social networks: identification and applications. Soc Netw Anal Min 6(1):66:1–66:17
Eltaher M, Lee J (2015) User profiling of Flickr: integrating multiple types of features for gender classification. J Adv Inf Technol 6(2):84–87
Figueiredo F, Ribeiro B, Almeida JM, Faloutsos C (2016) TribeFlow: mining and predicting user trajectories. In: Proceedings of the 25th international conference on world wide web, pp 695–706
Finkel JR, Grenager T, Manning C (2005) Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of the 43rd annual meeting on association for computational linguistics, pp 363–370
Ghahramani Z, Jordan MI (1997) Factorial hidden Markov models. Mach Learn 29(2–3):245–273
Hammersley JM, Clifford P (1971) Markov fields on finite graphs and lattices
Hu J, Zeng HJ, Li H, Niu C, Chen Z (2007) Demographic prediction based on user's browsing behavior. In: Proceedings of the 16th international conference on world wide web, pp 151–160
Ikeda K, Hattori G, Ono C, Asoh H, Higashino T (2013) Twitter user profiling based on text and community mining for market analysis. Knowl Based Syst 51(1):35–47
Joseph K, Wei W, Carley KM (2016) Exploring patterns of identity usage in tweets: a new problem, solution and case study. In: Proceedings of the 25th international conference on world wide web, pp 401–412
Kristjansson T, Culotta A, Viola P, McCallum A (2004) Interactive information extraction with constrained conditional random fields. In: Proceedings of the 19th national conference on artificial intelligence, pp 412–418
Krulwich B (1997) Lifestyle finder: intelligent user profiling using large-scale demographic data. AI Mag 18(2):37–45
Lafferty JD, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th international conference on machine learning, pp 282–289
Li R, Wang S, Deng H, Wang R, Chang KCC (2012) Towards social user profiling: unified and discriminative influence model for inferring home locations. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1023–1031
Li J, Ritter A, Hovy E (2014) Weakly supervised user profile extraction from Twitter. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, pp 165–174
Makazhanov A, Rafiei D, Waqar M (2014) Predicting political preference of Twitter users. Soc Netw Anal Min 4(1):193:1–193:15
McCallum A, Freitag D, Pereira FCN (2000) Maximum entropy Markov models for information extraction and segmentation. In: Proceedings of the 17th international conference on machine learning, pp 591–598
Michelson M, Knoblock C (2007) Unsupervised information extraction from unstructured, ungrammatical data sources on the world wide web. Int J Doc Anal Recogn 10(3):211–226
Pazzani M, Billsus D (1997) Learning and revising user profiles: the identification of interesting web sites. Mach Learn 27(3):313–331
Pedro JS, Siersdorfer S, Sanderson M (2011) Content redundancy in YouTube and its application to video tagging. ACM Trans Inf Syst 29(3):13:1–13:31
Richardson M, Domingos P (2006) Markov logic networks. Mach Learn 62(1–2):107–136
Ritze D, Lehmberg O, Oulabi Y, Bizer C (2016) Profiling the potential of web tables for augmenting cross-domain knowledge bases. In: Proceedings of the 25th international conference on world wide web, pp 251–261
Sarawagi S, Cohen WW (2004) Semi-Markov conditional random fields for information extraction. In: Proceedings of the 17th neural information processing systems, pp 1185–1192
Sarraute C, Brea J, Burroni J, Blanc P (2015) Inference of demographic attributes based on mobile phone usage patterns and social network topology. Soc Netw Anal Min 5(1):39:1–39:18
Soltysiak SJ, Crabtree IB (1998) Automatic learning of user profiles—towards the personalisation of agent services. BT Technol J 16(3):110–117
Szell M, Thurner S (2012) How women organize social networks different from men. ArXiv preprint arXiv:1205.4683
Tang J, Hong M, Li J, Liang B (2006) Tree-structured conditional random fields for semantic annotation. In: Proceedings of the 5th international conference on the semantic web, pp 640–653
Tang J, Hong M, Zhang D, Liang B, Li J (2007a) Emerging technologies of text mining: techniques and applications. Chap. Information extraction: methodologies and applications, pp 1–33. Idea Group Inc.
Tang J, Zhang D, Yao L (2007b) Social network extraction of academic researchers. In: Proceedings of the 7th IEEE international conference on data mining, pp 292–301
Tang J, Zhang J, Yao L, Li J, Zhang L, Su Z (2008) Arnetminer: extraction and mining of academic social networks. In: Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining, pp 990–998
Tang J, Yao L, Zhang D, Zhang J (2010) A combination approach to web user profiling. ACM Trans Knowl Discov Data 5(1):2:1–2:44
Tang W, Zhuang H, Tang J (2011a) Learning to infer social ties in large networks. In: ECML/PKDD'11, pp 381–397
Tang C, Ross K, Saxena N, Chen R (2011b) What's in a name: a study of names, gender inference, and gender behavior in Facebook. In: Proceedings of the 16th international conference on database systems for advanced applications, pp 344–356
Tang J, Fang Z, Sun J (2013) Incorporating social context and domain knowledge for entity recognition. In: Proceedings of the 24th international conference on world wide web, pp 517–526
Tang J, Lou T, Kleinberg J, Wu S (2016) Transfer learning to infer social ties across heterogeneous networks. ACM Trans Inf Syst 34(2):7:1–7:43
Weninger T, Han J (2013) Exploring structure and content on the web: extraction and integration of the semi-structured web. In:

Proceedings of the 6th ACM international conference on web search and data mining, pp 779–780

Weninger T, Hsu WH, Han J (2010) CETR: content extraction via tag ratios. In: Proceedings of the 19th international conference on world wide web, pp 971–980

Wu S, Liu J, Fan J (2015) Automatic web content extraction by combination of learning and grouping. In: Proceedings of the 24th international conference on world wide web, pp 1264–1274

Wu L, Ge Y, Liu Q, Chen E, Long B, Huang Z (2016) Modeling users' preferences and social links in social networking services: a joint-evolving perspective. In: Proceedings of the 30th AAAI conference on artificial intelligence, pp 279–286

Yedidia JS, Freeman WT, Weiss Y (2000) Generalized belief propagation. In: Proceedings of the 13th neural information processing systems, pp 689–695

Yu K, Guan G, Zhou M (2005) Resume information extraction with cascaded hybrid model. In: Proceedings of the 43rd annual meeting on association for computational linguistics, pp 499–506