

¹A Unified Approach to Researcher Profiling

Limin Yao, Jie Tang, and Juanzi Li

Department of Computer Science and Technology, Tsinghua University, China
{ylm,tangjie,ljz}@keg.cs.tsinghua.edu.cn

Abstract

This paper addresses the issue of researcher profiling. By researcher profiling, we mean building a semantic profile for an academic researcher, by identifying and annotating information from the Web. Previously, person profile annotation was often undertaken separately in an ad-hoc fashion. This paper first gives a formalization of the entire problem and proposes a unified approach to perform the task using Conditional Random Fields (CRF). The paper shows that with introduction of a set of tags, most of the annotation tasks can be performed within this approach. Experiments show that significant improvements over the separated method can be obtained, because the subtasks of annotation are interdependent and should be performed together. The method has been applied to expert finding. Experimental results show that the performance of expert finding can be significantly improved by using the profiling method.

1. Introduction

Profiling of a given person is the process of obtaining the values associated with the different properties that constitute the person model.

Traditionally, person profile annotation is viewed as an engineering issue and is conducted manually or undertaken separately in a more or less ad-hoc manner. For example, in web-based social networks such as MySpace.com and YouTube.com, the user has to enter the profile by her/him-self. Recently, a few research efforts have been made to automatically build the person profile using semantic annotation technologies [1][20]. However, existing methods use predefined rules or specific machine learning models to annotate different types of information in a separated fashion. This paper aims to conduct a thorough investigation on this issue. We focus on how to build a profile for a researcher using a unified approach by incorporating dependencies between different types of information.

1.1. Motivating Example

We begin by illustrating the problem with an example drawn from a real application of extracting researcher profiles in our developed system ArnetMiner (<http://www.arnetminer.org/>), which aims to build a semantic-based social network. Specifically, we define a researcher profile ontology, which include basic information (e.g. photo, affiliation, and position), contact information (e.g. address, email, and telephone), educational history (e.g. graduated university and major), and publications. For each researcher, we intend to create a profile based on the ontology by extracting the profile information from his/her homepage or Web pages introducing him/her. Figure 1 shows a researcher's homepage. It includes typical information in a researcher profile. The top section includes a photo, two addresses, and an email address; the middle section describes the educational history of the researcher; the bottom section provides the position and affiliation information. The ideal annotation result is shown in the right part of Figure 1.

Manually annotating the profile for each researcher is obviously tedious and time consuming. Recent work has shown the feasibility and promise of automatic extraction of semantic data from the Web, and it is possible to use the techniques to extract the profile.

However, most of the existing methods employed a predefined rule or a specific machine learning model to identify each type of information independently. It is *highly ineffective* to use the separated method to do researcher profiling due to the natural disadvantages of the method: (1) For each property in the profile, one has to define a specific rule or machine learning model. Therefore, there may be many different rules/models, which are difficult to maintain; (2) The separated rules/models cannot take advantage of dependencies across the different properties. The properties are (sometimes even strongly) dependent with each other. For example, *Name* can help recognition of the *Photo*. *Phdmajor* (PhD major) and *Phduniv* (PhD Univ.) are

¹ The work is supported by the National Natural Science Foundation of China under Grant No. 90604025.

We also found that 74.32% of the Web pages related to the 1K researchers are their homepages and the rest are pages introducing them. Characteristics of the two types of pages differ significantly from each other.

We analyzed the content of the Web pages, and found that 45.95% of the profile properties are presented in tables or lists on the Web pages and 54.05% of the profile properties are presented in natural language. This also indicates that a method without making full use of the global context information in the Web page would be ineffective to resolve the problem.

Statistical study also unveils that (strong) dependencies exist between different types of properties. For example in our data, there are 3842 cases (12.98%) that the annotation labels of the tokens require the annotation results of the other tokens. An ideal profiling model should consider processing all the subtasks together.

We do not take into consideration of extraction of publications. Instead, we integrate the publication data from existing online data source. We chose DBLP bibliography (dblp.uni-trier.de/), which is one of the best formatted and organized bibliography datasets. For integrating the researcher profiles and the publications data, we use researcher names and the author names as the identifier. The method has the name ambiguity problem (different researchers share the same name). We adapted a name-reconciliation algorithm [7], to resolve ambiguous names in our datasets. The algorithm employs a rigorous form of semantic similarity by gleaning the context associated with a researcher/author and based on a predefined threshold to reconcile the two names.

3. Our Approach

3.1. Process

There are three steps: relevant page finding, preprocessing, and tagging. In relevant page finding, given a researcher name, we first get a list of web pages by a web search engine (i.e. Google) and then identify the homepage or introducing page using a classifier. We view the URL of the identified web page as the value of the *Homepage* property in the profile.

In preprocessing, (A) we segment the text into tokens and (B) we assign possible tags to each token. The tokens form the basic units and the pages form the sequences of units in the tagging problem. In tagging, given a sequence of units, we determine the most likely corresponding sequence of tags using a trained tagging model. (The tags correspond to the properties implicit in Figure 2.)

(A). We identify tokens in the Web page heuristically. We define five types of tokens: ‘standard word’, ‘special word’, ‘<image>’ token, term, and punctuation mark. Standard words are unigram words in natural language. Special words [14] include email address, IP address, URL, date, number, percentage, words containing special symbols (e.g. ‘Ph.D.’, ‘Prof.’), unnecessary tokens (e.g. ‘===’ and ‘###’), etc. We identify special words using regular expressions. ‘<image>’ tokens are ‘<image>’ tags in the HTML file. We identify them by parsing the HTML file. Terms are base noun phrases extracted from the Web pages. We employed the methods proposed in [19]. Punctuation marks include period, question, and exclamation mark.

(B). We assign tags to each token based on their corresponding type. For standard word, we assign all possible tags. For special word, we assign tags: *Position*, *Affiliation*, *Email*, *Address*, *Phone*, *Fax*, and *Bsdate*, *Msdate*, and *Phddate*. For ‘<image>’ token, we assign two tags: *Photo* and *Email* (it is likely that an email address is shown as an image). For term token, we assign *Position*, *Affiliation*, *Address*, *Bsmajor*, *Msmajor*, *Phdmajor*, *Bsuniv*, *Msuniv*, and *Phduniv*. In this way, each token can be assigned several possible tags. Using the tags, we can perform most of the profiling processing (conducting 16 subtasks implicit in Figure 2 and covering 95.71% of the property values in the selected Web pages).

We do not conduct research interest annotation, although we could do it in theory. There are two reasons: first, we observed only one fifth of researchers provide the research interests on their homepages (21.31% of homepages in our data sets have research interests); second, research interests are usually implied by other profile properties, e.g., papers published by the researcher or research projects he/she is involved in. See [17] for details.

3.2. Conditional Random Fields

We employ Conditional Random Fields (CRF) as the tagging model. CRF is a conditional probability distribution of a sequence of labels given a sequence of observations, represented as $P(Y|X)$, where X denotes the observation sequence and Y the label sequence [12]. All components Y_i of Y are assumed to range over a finite label alphabet Y (as the properties defined in Figure 2). The conditional probability is formulized as:

$$p(y|x) = \frac{1}{Z(x)} \exp \left(\sum_{e \in E, j} \lambda_j t_j(e, y|_e, x) + \sum_{v \in V, k} \mu_k s_k(v, y|_v, x) \right)$$

where x is a data sequence, y is a label sequence, and $y|_e$ and $y|_v$ are the set of components of y associated with edge e and vertex v in the linear chain respectively; t_j and s_k are feature functions; parameters

λ_j and μ_k are coefficients corresponding to the feature functions t_j and s_k respectively, and are to be estimated from the training data; $Z(x)$ is the normalization factor.

In tagging, the model is used to find the sequence of tags Y^* with the highest likelihood $Y^* = \max_Y P(Y|X)$, using the Viterbi algorithm.

In training, the CRF model is built with labeled data and by means of an iterative algorithm based on Maximum Likelihood Estimation.

3.3. Features

For each token unit, three types of features are defined: content features, pattern features, and term features.

1. Content Features

For a standard word, the content features include:

(1) Word features. Whether the current token is a standard word.

(2) Morphological features. The morphology of the current token, e.g. whether the token is capitalized.

For a '<image>' token, the content features include:

(1) Image size. The size of the current image.

(2) Image height/width ratio. The ratio of the height to the width of the current image. The ratio of a person photo is likely to be greater than 1.0.

(3) Image format. The format of the image (e.g. "JPG", "BMP").

(4) Image color. The number of the "unique color" used in the image and the number of bits used for per pixel (e.g. 32, 24, 16, 8, and 1).

(5) Face recognition. Whether the current image contains a person face. We used a face recognition tool (<http://opencvlibrary.sf.net>) to detect the person face.

(6) Image filename. Whether the image filename (partially) contains the researcher name.

(7) Image "ALT". Whether the "alt" attribute of the "<image>" tag (partially) contains the researcher name.

(8) Image positive keywords. Whether the image filename contains positive keywords like "myself".

(9) Image negative keywords. Whether the image filename contains negative keywords like "logo".

2. Pattern Features

Pattern features are defined for each token.

(1) Positive words. Whether the current token contains positive *Fax/Phone* keywords like "Fax:", "Phone:", positive *Position* keywords like "Manager".

(2) Special tokens. Whether the current token is a special word.

3. Term Features

Term features are defined only for term token.

(1) Term features. Whether the token unit is a term.

(2) Dictionary features. Whether the term is included in a dictionary.

We can easily incorporate these features into our model by defining Boolean-valued feature functions. Finally, two sets of features are defined in the CRF model: transition features and state features. For example, a transition feature $y_{i-1}=y', y_i=y$ implies that if the current tag is y and the previous tag is y' , then the value is true; otherwise false. The state feature $w_i=w, y_i=y$ implies that if the token is w and the current tag is y , then the feature value is true; otherwise false. In total, 108,409 features were used in our experiments.

3.4. Baseline Methods

We can consider two baseline methods, namely rule based and classification based approach. Both of the approaches perform profile annotation with several passes, taking the raw text as input and conduct annotation independently for each property.

3.5. Advantages of Our Approach

Our method offers some advantages. (1) The annotations of different properties are interdependent. The separated method cannot simultaneously perform the annotation tasks. In contrast, our method can effectively overcome the drawback by employing a unified framework and achieve better performance. (2) There are many specific properties in the profile. If one defines a specialized model or rule to handle each of the cases, the number of models needed will be large and difficult to maintain. Our method formalizes all the tasks as assigning different types of tags and trains a unified model to tackle all the subtasks.

4. Experimental Results

In this section, we report empirical results by applying the proposed unified approach to extract the researcher profile. We utilized the tool KEG_CRF (http://keg.cs.tsinghua.edu.cn/persons/tj/software/KEG_CRF) with all the default settings.

4.1. Experimental Setup

4.1.1. Datasets. We randomly chose in total 1K researcher names from ArnetMiner. We used the method described in Section 3.1 to find the researchers' homepages or introducing pages (with F1-score of 92.39% for classification). If the method cannot find a Web page for a researcher, we remove the researcher name from the data set. Finally 898 Web pages were obtained.

Seven human annotators conducted annotation on the Web pages. A spec was created to guide the annotation process. All profile properties were labeled.

For disagreements in the annotation, we conducted “majority voting”.

We produced statistics on the data set. In summary, 86.41% of the Web pages contain at least five properties and 96.44% contain four. We also found that some properties (e.g. *Position* and *Affiliation*) tend to appear more than once on a page. We omit the details due to space limitation.

4.1.2. Evaluation Measures. In the experiments, we conducted evaluations in terms of precision, recall, and F1-measure (for definitions of the measures, see [18]). By comparison of the other work, we also give statistical significance estimates using Sign Test [9].

4.1.3. Implementation of baseline methods. We use the rule learning based and the classification based approach as baselines.

For the rule learning based approach, we employed Amilcare [4], which utilizes a rule induction algorithm LP² to learn annotation rules from the training data.

For the classification based approach, we employed Support Vector Machines (SVM) [5]. In SVM we used the same features as that in our unified model. We selected “one-versus-all” strategy, taking one property as positive examples and others as negative ones to train a classifier. 16 classifiers were trained, corresponding to 16 subtasks.

To test how the dependencies between different types of properties affect the profiling performance, we also conducted experiments using the unified model by removing the transition features (Unified_NT).

4.2. Researcher Profiling Results

4.2.1. Results. We evaluated the performance of all the methods on the dataset. Table 1 lists the five-fold cross-validation results. Our method outperforms the baseline methods. It can be also seen that the performance of the unified method decreases when removing the transition features (Unified_NT). (We should note that for *Photo*, the rule-based method cannot incorporate content features of images. Thus the performance is worse than other methods.)

We conducted sign tests for each annotation subtask, which show that all the improvements of Unified over Amilcare, SVM, and Unified_NT are statistically significant ($p \ll 0.01$).

4.2.2. Contribution of features. We investigated the contribution of each feature type in profile annotation. We employed only content features, content+term features, content+pattern features, and all features to train the models and conducted the annotation.

Figure 3 shows the macro-average F1-score of profile annotation with different feature types. We see

that solely using one type of feature alone cannot accomplish accurate annotation.

Table 1. F1-measure of all the methods (%)

Profiling Task	Unified	Unified_NT	SVM	Amilcare
Photo	89.11	88.64	88.86	31.62
Position	69.44	64.70	64.68	56.48
Affiliation	83.52	72.16	73.86	46.65
Phone	91.10	78.72	79.71	83.33
Fax	90.83	64.28	64.17	86.88
Email	80.35	75.47	79.37	78.70
Address	86.34	75.15	77.04	66.24
Bsuniv	67.38	57.56	59.54	47.17
Bsmajor	64.20	59.18	60.75	58.67
Bsdate	53.49	40.59	28.49	52.34
Msuniv	57.55	47.49	49.78	45.00
Msmajor	63.35	61.92	62.10	57.14
Msddate	48.96	41.27	30.07	56.00
Phduniv	63.73	53.11	57.01	59.42
Phdmajor	67.92	59.30	59.67	57.93
Phddate	57.75	42.49	41.44	61.19
Overall	83.37	72.09	73.57	62.30

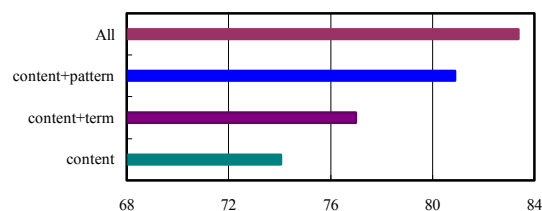


Figure 3. Contribution of features

4.2.3. Discussions. Our method outperforms the rule based and SVM based methods in almost all the subtasks, especially in the subtasks that have strong dependencies with each other, for example, the annotations of *Affiliation* and *Position* and the nine subtasks in educational history.

The baseline methods suffered from ignorance of dependencies between the subtasks. For example, there are 1460 cases (19.31%) of *Affiliation* need to use the results of *Position*; 17.66% of the educational history information (e.g. *Phdmajor* and *Phduniv*) needs use dependencies with each other. However, the baseline methods cannot make use of the dependencies, as it conducts all the subtasks from raw input data.

Our method benefits from the ability of modeling dependencies between subtasks. We see from Table 1 that by leveraging the dependencies, our method can outperform the method without using dependencies (Unified_NT) by 10.28% in terms of F1-measure.

Here we use an example to show the advantage of our methods compared with the methods without utilizing dependencies. The input text is: (“<tag>” and “</tag>” are labeled tags)

“He received a B.A. in <bsmajor>Philosophy</bsmajor> from <bsuniv>Oberlin College</bsuniv> in <bsdate>1984</bsdate>”, and a

Ph.D. from the Department of <phdmajor>Computer Science</phdmajor> at the <phduniv>University of Maryland</phduniv> in <phddate>1992</phddate>.”

With profiling by Amilcare, we obtain:

“He received a B.A. in <bsmajor>Philosophy</bsmajor> from <bsuniv>Oberlin College</bsuniv> in <bsdate>1984</bsdate>, and a Ph.D. from the Department of <phdmajor>Computer Science</phdmajor> at the **University of Maryland** in <phddate>1992</phddate>.”

With profiling by SVM, we obtain:

“He received a B.A. in <bsmajor>Philosophy</bsmajor> from <bsuniv>Oberlin College</bsuniv> in <phddate>1984</phddate>, and a Ph.D. from the Department of <phdmajor>Computer Science</phdmajor> at the <phduniv>University of Maryland</phduniv> in **1992**.”

With profiling by Unified_NT, we obtain:

“He received a B.A. in **Philosophy** from <bsuniv>Oberlin College</bsuniv> in <bsdate>1984</bsdate>, and a Ph.D. from the Department of <phdmajor>Computer Science</phdmajor> at the **University of Maryland** in **1992**.”

With profiling by our method, we obtain:

“He received a B.A. in <bsmajor>Philosophy</bsmajor> from <bsuniv>Oberlin College</bsuniv> in <bsdate>1984</bsdate>, and a Ph.D. from the Department of <phdmajor>Computer Science</phdmajor> at the <phduniv>University of Maryland</phduniv> in <phddate>1992</phddate>.”

Amilcare can annotate some properties correctly. However, it does not recognize “University of Maryland” as *Phduniv*.

SVM can annotate some of the properties correctly, as well. For instance, it can detect the *Bsmajor* and *Bsuniv*. However, it mistakenly identified “1984” as the *Phddate* and cannot identify the *Phddate* “1992”.

The Unified_NT method can perform all the tasks simultaneously. However, as it does not make use of the dependencies between subtasks, it cannot identify the *Bsmajor* “Philosophy”, *Phduniv* “University of Maryland”, and the *Phddate* “1992”.

Our method can take advantage of the dependencies among the subtasks and thus correct 9.95% of the errors that Unified_NT cannot handle. Similarly, our method can correct 8.41% of the errors that SVM cannot handle and 19.07% of the errors that Amilcare cannot handle.

Although we conducted error analysis on the results, we omit the details here due to space limitation.

4.3. Expert Finding Experiments

To further evaluate the effectiveness of our method, we applied it to expert finding. The task of expert finding is to identify persons with some expertise or experience on a specific topic in a social network. One common practice of finding information using social networks is to ask an expert and thus expert finding plays an important role in social networks. We designed this experiment to see whether the annotated

profile information can improve the performance of expert finding.

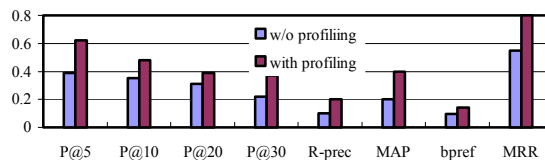


Figure 4. Performances of expert finding

We selected 12 topics for finding experts from ArnetMiner. See [21] for details. We conducted evaluation in terms of P@5, P@10, P@20, P@30, R-prec, mean average precision (*MAP*), *bpref*, and mean reciprocal rank (*MRR*) [6].

Figure 4 shows the results of expert finding. We see that significant improvements can be obtained using the annotated profile information.

5. Related Work

Extracting profile information of a person from the Web can benefit many Web applications. Several research efforts have been made so far. For example, Yu et al. propose a cascaded information extraction framework for identifying personal information from resumes [20]. The Artequakt system can automatically extract knowledge of artists from the Web [1]. However, most of the previous works view the profile annotation as several separated issues and conduct annotation in a more or less ad-hoc manner. To the best of our knowledge, no previous work has been done on researcher profiling using a unified approach.

Considerable efforts have been placed on extraction of contact information from emails or the Web. For example, Kristjansson et al. developed an interactive information extraction system to assist the user to populate the fields of a contact database [11]. Balog and Rijke employed heuristic rules to extract contact information from emails [2]. See also [15]. Contact information extraction is only a subtask of profile annotation. It differs from profile annotation significantly in nature.

Many annotation systems have been developed by casting the annotation task as rule induction and classification. For example, Ciravegna et al. propose a rule learning algorithm, called LP² [4]. The rule induction based method can achieve good results on the template based web pages. However, it cannot utilize dependencies across targeted instances.

SCORE Enhancement Engine (SEE) supports web page annotation using classification model [10]. The classification based method obtains good results on many annotation tasks. However, it cannot also use the dependencies across different targeted instances.

Sequential labeling describes dependencies between targeted instances to improve the annotation accuracy. However, limited research has been done using the sequential labeling method in semantic annotation.

Many information extraction models have been proposed. Hidden Markov Model [8], Maximum Entropy Markov Model [13], Conditional Random Fields [12], Support Vector Machines [5] are widely used extraction models. See [16] for an overview.

6. Conclusion

In this paper, we have investigated the problem of researcher profile annotation, an important issue for building semantic-based social networks. We have defined the problem as a task consisting of 19 subtasks. We have then proposed a unified approach to perform the task, specifically to treat profile annotation as assigning tags to the tokens in the web pages. Experimental results show that our approach outperforms the independent baseline methods for profile annotation. When applying the profile extraction method to expert finding, we observed significant improvements. As future work, we plan to make further improvement on the profiling accuracy. We also plan to utilize the researcher profiles to help other applications, for example name disambiguation.

7. References

- [1] H. Alani, S. Kim, D. Millard, M. Weal, W. Hall, P. Lewis, and N. Shadbolt. Automatic Ontology-Based Knowledge Extraction from Web Documents. *IEEE Intelligent Systems*. 2003, 18(1):14-21.
- [2] K. Balog and M. Rijke. Finding Experts and Their Details in E-mail Corpora. In *Proc. of WWW2006*.
- [3] D. Brickley and L. Miller. FOAF Vocabulary Specification, *Namespace Document*, September 2, 2004. <http://xmlns.com/foaf/0.1/>.
- [4] F. Ciravegna. (LP)², an Adaptive Algorithm for Information Extraction from Web-related Texts. In *Proc. of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, Seattle, USA, August 2001.
- [5] C. Cortes and V. Vapnik. Support-Vector Networks. *Machine Learning* 20, 1995:273-297.
- [6] N. Craswell, A. de Vries, and I. Soboroff. Overview of the Trec-2005 Enterprise Track. *TREC 2005 Conference Notebook*. pp.199-205.
- [7] X. Dong, A. Halevy, and J. Madhavan. Reference Reconciliation in Complex Information Spaces. In *Proc.SIGMOD2005*.
- [8] Z. Ghahramani and M. Jordan. Factorial Hidden Markov Models. *Machine Learning*, 1997, 29:245-273.
- [9] L. Gillick and S. Cox. Some Statistical Issues in the Comparison of Speech Recognition Algorithms. In *Proc. of ICASSP1989*.
- [10] B. Hammond, A. Sheth, and K. Kochut. Semantic Enhancement Engine: a Modular Document Enhancement Platform for Semantic Applications over Heterogeneous Content, in Real World Semantic Web Applications. IOS Press, 2002. pp. 29-49.
- [11] T. Kristjansson, A. Culotta, P. Viola, and A. McCallum. Interactive Information Extraction with Constrained Conditional Random Fields. In *Proc. of AAAI2004*.
- [12] J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML2001*.
- [13] A. McCallum, D. Freitag, and F. Pereira. Maximum Entropy Markov Models for Information Extraction and Segmentation. In *Proc. of the ICML2000*.
- [14] R. Sproat, A. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards. Normalization of Non-Standard Words, WS'99 Final Report.
- [15] J. Tang, H. Li, Y. Cao, and Z. Tang. Email Data Cleaning. In *Proc. of KDD'2005*. pp. 489-499
- [16] J. Tang, M. Hong, D. Zhang, B. Liang, and J. Li. Information Extraction: Methodologies and Applications. In the book of *Emerging Technologies of Text Mining: Techniques and Applications*, 2007. (to appear)
- [17] J. Tang, M. Hong, J. Zhang, B. Liang, L. Yao, and J. Li. ArnetMiner: Toward Building and Mining Social Networks. (Demo) In *Proc. of SIGKDD2007*.
- [18] C.J. van Rijsbergen. Information Retrieval. 1979.
- [19] E. Xun, C. Huang, and M. Zhou. A Unified Statistical Model for the Identification of English baseNP. In *Proc. of ACL2000*.
- [20] K. Yu, G. Guan, and M. Zhou. Resume Information Extraction with Cascaded Hybrid Model. In *Proc. of ACL2005*.
- [21] J. Zhang, J. Tang, and J. Li. Expert Finding in a Social Network. In *Proc. of DASFAA'2007*. pp. 1066-1069.