An Unsupervised Method for Author Extraction from Web Pages Containing User-Generated Content*

Jing Liu†, Xinying Song‡, Jingtian Jiang§ and Chin-Yew Lin§
† Harbin Institute of Technology, Harbin 150001, P.R.China
‡ Microsoft Research, Redmond, WA 98052, USA
§ Microsoft Research Asia, Beijing 100080, P.R.China
jliu@ir.hit.edu.cn, {xinson, jiji, cyl}@microsoft.com

ABSTRACT

In this paper, we address the problem of author extraction (AE) from user generated content (UGC) pages. Most existing solutions for web information extraction, including AE, adopt supervised approaches, which require expensive manual annotation. We propose a novel unsupervised approach for automatically collecting and labeling training data based on two key observations of author names: (1) people tend to use a single name across sites if their preferred names are available; (2) people tend to create unique usernames to easily distinguish themselves from others, e.g. `travelbug61`. Our AE solution only requires features extracted from a single UGC page instead of relying on clues from multiple UGC pages. We conducted extensive experiments. (1) The evaluation of automatically labeled author field data shows 95.0% precision. (2) Our method achieves an F1 score of 96.1%, which significantly outperforms a state-of-the-art supervised approach [10] with single page features (F1: 68.4%) and has a comparable performance to its multiple page solution (F1: 95.4%). (3) We also examine the robustness of our approach on various UGC pages from forums and review sites, and achieve promising results as well.

Categories and Subject Descriptors

H.3.m [**Information Storage and Retrieval**]: Miscellaneous - Data Extraction; Web

General Terms

Algorithms, Performance, Experimentation

Keywords

Author extraction, unsupervised approach

1. INTRODUCTION

As the web becomes more and more social, social networks such as Twitter, Facebook, forums, blogs, and so on are capturing a tremendous amount of user interaction information in the form of UGC. Studying this rich user interaction information in UGC could lead to the design of innovative social-enriched services and new insights into how people interact in the real world. For example, Bing recently released “People Who Know” and “Friends Who Might Know” features, which recommend topic experts and friends relevant to user tasks. Liu et al. [4] explored expert identification from question answering sites. Gruhl et al. [3] studied the problem of people's influence estimation in the blogsphere. Abel et al. [1] proposed aggregating people's public data from different UGC sites to address the cold-start problem in recommender systems. In addition, a lot of research into community structure analysis has been conducted on people networks constructed from UGC sites [6].

To enable services or research mentioned above, it is necessary to have a comprehensive people (or user) database that stores and indexes relevant people information in a targeted domain. For example, Bing's “Friends Who Might Know” utilizes the people database from Facebook and its “People Who Know” relies on people information from UGC. In general, it is not difficult to build a site-specific people database with manually written information extraction rules. However, scaling site-specific solutions to the web scale is a challenge. To extract user information from publicly available UGC sites typically involves the following two steps: 1. user post record extraction given a web page containing UGC data, for example, a product review page on Amazon.com; 2. author extraction from a user post record. A great deal of research effort has been devoted to extracting user post records from UGC pages [7, 10]. Current state-of-the-art systems [7, 10] show satisfactory results on post record extraction. However, less attention has been paid to addressing the author extraction (AE) problem. The most relevant work on AE has been conducted by Yang et al. in [10]. Their Markov Logic Network (MLN) based methods achieve an F1 score of 68.4% using features from a single UGC page and an F1 score of 95.4% when multiple UGC pages of a site are used. Though they have shown that effective author extraction is possible by incorporating features from multiple UGC pages, methods relying on single page features still leave much to be desired. An effective single page AE solution does not require site level knowledge and could be incorporated in online web crawlers to extract author information on the fly. In this paper, we focus on solving the author extraction task by only utilizing features from a single UGC page. As we will show in Sec. 4, our method achieves

*This research was conducted while Jing Liu was a visiting student at Microsoft Research Asia.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

an F1 score of 96.1% which significantly outperforms Yang et al.'s MLN-based single page solution (F1 score: 68.4%) and has a comparable performance to their multiple page solution (F1 score: 95.4%).

In addition to only considering single page features, our method automatically creates an AE training corpus given a set of seed author names that are statistically rare character strings, e.g. `travelbug61`, when compared to ordinary English words. It then trains a SVM author name classifier over the training corpus to detect the author name field in a user post record. Most existing solutions for web information extraction [12, 5, 10] adopt supervised approaches. Supervised learning methods are very effective but require manually annotated training corpora that are labor-intensive and expensive to acquire. Our solution does not require manually annotated corpora and is based on two key observations of author names: (1) people tend to use the same author (or user) name across different sites and (2) people prefer choosing unique author names, e.g. `bennystar99`. These two observations ensure our method effectively and efficiently obtains and labels training data for learning classifiers.

Our paper is organized as follows: we first discuss how to automatically create training data for author name extraction in Sec. 2. We then introduce our author extraction method in Sec. 3. Comprehensive evaluations and comparisons to existing methods are detailed in Sec. 4. We conclude and propose possible future directions in Sec. 5.

2. AUTOMATIC ACQUISITION OF TRAINING DATA

We propose an unsupervised approach that automatically collects and labels training data for author extraction.

Our approach mainly includes two phases: (a) obtaining UGC page candidates and (b) labeling true UGC pages from the obtained pages and labeling author fields on the labeled UGC pages. The first phase starts by populating a set of seed author names from a few initial UGC sites. It is expected that the owners of the seed author names left their footprints on other UGC sites. Hence, we select the most unique author names from the seed set as queries to a commercial search engine to collect the pages where the seed author names appear. The new discovered pages may include UGC pages (blog comment pages, product review pages, etc.) and non-UGC pages (user profile pages, member list pages, thread index pages, etc.). In the second phase, we propose an MiBAT [7] based method which leverages the queries to automatically label true UGC pages and author fields for learning classifiers.

2.1 Obtaining UGC Page Candidates

It is expected that using seed author names as search queries can help us collect a web page set containing a certain number of UGC pages. However, in practice, author names can be composed of any words. Some author names (e.g. "`Blues`") are common words that could be used by many people for any purpose rather than only for author names. If such author names are submitted as queries to a commercial search engine, it is likely that the returned pages contain very few UGC pages. In contrast, it is intuitive that some unique author names (e.g. "`travelbug61`") can effectively and efficiently help us obtain UGC pages.

According to these observations, we define the n-gram probability of an author name to select the most unique author names as queries for collecting UGC page candidates. When people give their own author names, they would like to use some combinations of word sequences (one or more words) as their author names. The word sequences may present people's real names, birthdays, etc. The n-gram probability of an author name is defined as the n-gram probability of the word sequence of which the author name consists. The lower the n-gram probability of an author name is, the more likely that the author name is unique.

Since it is not allowed to have spaces in author names at many sites, we should first perform word breaking on author names before estimating the n-gram probabilities. The problem of word breaking on an author name can be formalized as follows:

$$\hat{s} = \arg\max\nolimits_{s \subseteq \Omega} P(s|a) = \arg\max\nolimits_{s \subseteq \Omega} P(a|s)P(s)$$

Here, $s = (w_1, w_2, w_3, \ldots, w_{|s|})$ is a segmentation of an author name a (without spaces) and the $\hat{s}$ is the objective segmentation, where $|s|$ denotes the number of words in the segmentation s. Let $|a|$ present the number of characters in a, the size of the set of all possible segmentations Ω is $2^{|a|-1}$. In addition, $P(a|s)$ and $P(s)$ are called the transformation and the segmentation prior model, respectively. In this paper, we use the word synchronous beam search approach proposed by [8] to to estimate $P(a|s)$ and $P(s)$ and use the web n-gram service provided by [9] to estimate the n-gram probability. In this paper, we use the 5-gram based on the title corpus provided by a web n-gram service.

2.2 Automatically Labeling Training Data

The obtained pages via seed author names contain UGC pages and non-UGC pages. Intuitively, a UGC page should contain a list of post records (criterion 1). Moreover, each post record should contain at least three data fields: author (criterion 2), posting time (criterion 3) and content (criterion 4). Based on this intuition, we propose an MiBAT [7] based method that leverages the queries triggering the obtained pages. We are thus able to identify true UGC pages among the obtained pages and label the author field on the identified UGC pages for learning classifiers.

Alg. 1 shows our approach to automatically labeling training data by considering criteria 1 ∼ 4. For each obtained page, we first use MiBAT [7] to detect and extract the main region, which consists of a list of records (criterion 1) containing a time field (criterion 2) (Line 3 ∼ 4). Every data record list is called a data region [7]. In our scenario, the main region is the data region that most likely contains UGC. MiBAT [7] is an unsupervised method that extracts post records from a given UGC page by leveraging posting time. It assumes that each post record contains a posting time field. In other words, given any web page, MiBAT can detect and extract the main region that consists of a list of records containing a time field.

Since UGC pages are usually generated from a database and represented via templates, a data field (e.g. author) may repetitively appear as one node in each data record. It is assumed that the DOM tree nodes in different records, which belong to the same field, can be aligned via tree alignment. As in [7, 11], we use a top-down tree alignment algorithm to align the DOM trees of all post records (Line 6). Here, a field is a set of aligned DOM tree nodes.

Next, we determine if there is one field containing one node that exactly matches the query triggering the page. If there is one such field, the field will be annotated as the author field (Line 7 ∼ 12). Otherwise, the field will be annotated as a non-author field. Here, a characteristic of the seed author names is leveraged again. Since the query has a low n-gram probability, it is unlikely that the query appears in other fields (e.g. title, content). This guarantees the high precision of author name field annotation.

There might be some list pages (e.g. forum thread list pages) in the set of obtained pages. They may match the first three criteria, but they are not UGC pages according to criterion 4. Usually, such list pages contain a field that (a) has the longest text length and (b) has a link. We use one heuristic to filter out such pages (Line 13 ∼ 15). Finally, we get the labeled training data for author name extraction (Line 17). Note that our approach is unsupervised, because there is no learning using human annotated labels, except for a few initial sites where human annotation is required to have a set of seed author names.

Algorithm 1 Automatically Labeling Training Data

```
Input: A set $S$ of web pages obtained by the queries (seed author names)
Output: A set $\Omega$ of labeled UGC web pages with an annotated author field for learning classifiers
 1: $\Omega \leftarrow \{\}$
 2: for each $(p_i, q_i)$ in $S$ do        ▷ $q_i$ is the query (author name) triggering the page $p_i$
 3:     $R \leftarrow$ MiBAT$(p_i)$   ▷ $R$ is the main region consisting of a list of records containing a time stamp
 4:     if $R$ is not empty then
 5:         $hasAuthor \leftarrow$ False
 6:         $F \leftarrow$ ExtractFieldsByTreeAlignment$(R)$   ▷ $F$ is a set of fields
 7:         for each $f_k$ in $F$ do                ▷ $f_k$ is a field
 8:             if $q_i$ in $f_k.nodes$ then   ▷ If one node of field $f_k$ exactly matches $q_i$ (author name)
 9:                 $f_k.label \leftarrow$ Author     ▷ Label author field
10:                 $hasAuthor \leftarrow$ True
11:             else
12:                 $f_k.label \leftarrow$ Non-Author
13:         $\hat{f} \leftarrow$ SelectLongestField$(F)$
14:         if $\hat{f}.hasLink \neq$ True and $hasAuthor$ = True then
15:             $p_i.label \leftarrow$ UGC              ▷ Label page
16:             $\Omega \leftarrow \Omega \cup (p_i, F)$
17: return $\Omega$
```

3. METHOD AND FEATURES

3.1 Method for Author Extraction

Following the standard solutions for web information extraction [12, 5, 10], we view author extraction as a classification task. As mentioned in Sec. 2.2, it is assumed that the DOM tree nodes in different records, which belong to the same field (e.g. author), can be aligned via tree alignment. Hence, we first use MiBAT [7] to extract post records from a given UGC page and use a top-down tree alignment algorithm [11] to align the DOM trees of all extracted records. Then, we do classification on each field to extract authors. In this paper, we use LibSVM [2] with the RBF kernel as a classifier. If there are multiple fields classified as author fields in one post record, we only keep the one with the maximum probability of being an author field.

3.2 Features for Author Extraction

Usually, an author field on a UGC page has its own typical layout. We propose four types of features for author extraction. Table 1 describes each type of main features.

- ***Visual features*** The text of an author field is usually short and emphasized by HTML tags <bold> or <strong>. Additionally, there is usually one author field in each record. These characteristics can be observed upon inception. Hence, we called these visual features.
- ***Text features*** Usually, author names are composed by out of vocabulary (OOV) terms. In our observation, author names often start with letters and end with digits. This guarantees the uniqueness of author names.
- ***Link features*** We observed that there are out links in author fields at many UGC sites. Those links should point to the author profile pages. Moreover, there are URL patterns that can be found in those links (e.g. http://www.physicsforums.com/member.php?u=\d+). Since many UGC sites use parameterized URLs.
- ***Context features*** In our observation, some tokens can be good indicators for author or non-author fields. Those tokens can appear as text prefix/suffix (e.g. "posted by" for an author field, "joined on" for a non-author field), HTML tag attributions ("authorname" for an author field, "date" for a non-author field) and URL ("showuser" for an author field, "viewforum" for a non-author field). However, it is hard to enumerate such tokens by hand. Fortunately, the tokens indicating different fields can be learned from the large amount of automatically labeled training data. The tokens indicating author or non-author fields can be learned by setting thresholds on (a) the probability (70%) that they appear related to an author or non-author field and (b) their frequency (10 times) in the automatically labeled training data.

Table 1: Main Features for Author Extraction

Type	Description
Visual Features	The ratio of nodes with a short text length (between 3 ∼ 20) in this field
	The maximum/average/variance text length among all the nodes in this field
	Whether this field is emphasized by HTML tagss
	The ratio of the number of aligned nodes in this field to the number of records in this page
Text Features	The ratio of OOV terms in the field
	The ratio of nodes with pure digits in this field
	The ratio of nodes with a pattern that starts with letters and ends with digits in this field
Link Features	The ratio of nodes containing links in this field
	Whether all the out links in this field point to pages within this pages web-site
	Whether all the out links in this field share the same URL pattern
	Whether all the out links in this field are the same
	The ratio of nodes with an out link containing the node text (A link pointing to an author profile page may contain author name)
Context Features	Whether there is author or non-author related text prefix/suffix (e.g. "posted by:", "joined on:")
	Whether there are author or non-author related HTML tag attributes (e.g. "postauthor", "date")
	Whether there are author or non-author related URL tokens (e.g. "members", "showthread")

4. EXPERIMENTS AND EVALUATION

In this section, we (1) evaluate the quality of the automatically obtained training data (Sec. 4.2), (2) test the effectiveness of our approach by comparing our approach with one state-of-the-art supervised approach [10] (Sec. 4.3), (3) test the scalability of our approach by applying it to various types of forum thread pages and review pages (Sec. 4.4).

4.1 Evaluation Metric

An author field in a post record is regarded as correct if it contains exactly the same text as the one (author name) within the manually annotated author field. We use this text-based judgement because there might be multiple acceptable author fields in one post record. We use standard precision and recall as evaluation metrics. Since the post records should first be extracted using MiBAT [7], there will be accumulated errors. Hence, there are two cases separately reported (1) all post records (AP for short) and (2) the correctly extracted post records (CP for short).

4.2 Training Data Evaluation

We collected data from 54 forums and extracted author names by manually writing a wrapper for each site. Then, we selected the most unique author names (top 1% from each site) as queries by measuring the n-gram probability of each author name. There were 23,395 selected queries, 852,261 unique URLs obtained from Google search result and 69,158 unique web sites. We randomly sampled and crawled at most 5 URLs from each web site. Finally, we had 117,062 pages from 56,479 web sites, among which there were 30,770 pages automatically labeled as UGC pages.

Table 2: Evaluation on automatic author field annotation

	Golden UGC pages		**Correct UGC pages**	
	AP	**CP**	**AP**	**CP**
Prec.	0.950	0.958	0.950	0.958
Rec.	0.738	0.934	0.931	0.958

To examine the quality of the automatically labeled training data, we randomly sampled and manually annotated 200 web pages. There were 76 true UGC pages, which mainly included forum thread pages, review pages, and comment pages. The non-UGC pages mainly included profile pages, thread index pages, and member list/ranking pages. Our method for labeling UGC pages showed 94.92% precision and 73.68% recall. We further evaluated the quality of automatically labeled author field on two sets of UGC pages: (a) manually annotated true UGC pages (named Golden UGC pages) and (b) automatically labeled UGC pages that are correct (named Correct UGC pages). Table 2 shows the satisfactory results for automatic author field annotation.

4.3 Comparison with a Supervised Method

In this section, we compare our approach with one state-of-the-art supervised approach [10] based on Markov logic networks (MLNs). We use the same data set as the one in [10]. There are 20 forum sites and 1,000 thread pages from each site in this data set. Table 3 shows the comparison results. In Table 3, MLNs-P means the supervised model in [10] using only single page features; MLNs-PPV means the supervised model in [10] using both single page features and site-level features. Note that our approach only uses single page features. The results show that our approach is very effective: (1) our approach outperforms the supervised model in [10] only using single page features; (2) our approach produces a comparable performance to the supervised model in [10] using both single page features and site-level features.

Table 3: Comparison with Yang et al. [10]

Model	Ours	MLNs-P	MLNs-PPV
Prec. (AP)	**0.967**	0.751	0.939
Rec. (AP)	0.955	0.585	**0.969**
F1 (AP)	**0.961**	0.658	0.954

4.4 Experiments on Forum and Review Data

We created a data set consisting of 197 forum pages from 141 forum sites and 234 review pages from 14 popular product review sites. The left part of Table 4 shows the experimental results of our approach with all features. It shows that our approach is effective and scalable to various types of UGC sites. As mentioned in Sec. 3.2, our approach mainly benefits from the ***context features***, which are learned from a large amount of automatically obtained and labeled training data. From Table 4, we can see that our classification model without the learned ***context features*** shows significant performance degradation. This demonstrates the importance of context features and the ability of our method to learn them automatically.

Table 4: Experiments on forum threads and reviews

	All features		**Leaving out context features**	
	forum	review	forum	review
Prec.(AP)	0.878	0.876	0.854	0.813
Rec.(AP)	0.824	0.751	0.801	0.698
Prec./Rec.(CP)	0.911	0.902	0.886	0.838

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose an unsupervised author extraction method based on features of a single UGC page and show how to automatically obtain and label training data for learning classifiers based on two key observations of author name composition. The experimental results show that our method is very effective, achieving an F1 score of 96.1%, which is significantly better than the state-of-the-art results reported by Yang et al. in a similar setting (F1 of 65.8%) and is comparable to their multiple page settings (F1 of 95.4%). In the future, we would like to explore the possibility of adding an iterative phase to our method by using newly harvested rare author names to create a large people database and extending our method to the extraction of other types of semi-structured data on the web.

6. REFERENCES

[1] F. Abel, N. Henze, E. Herder, and D. Krause. Interweaving public user profiles on the web. *UMAP*, 2010.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2011.

[3] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW*, 2004.

[4] J. Liu, Y. Song, and C. Lin. Competition-based user expertise score estimation. In *SIGIR*, 2011.

[5] P. Luo, F. Lin, Y. Xiong, Y. Zhao, and Z. Shi. Towards combining web classification and web information extraction: a case study. In *SIGKDD*, 2009.

[6] M. Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 2011.

[7] X. Song, J. Liu, Y. Cao, C. Lin, and H. Hon. Automatic extraction of web data records containing user-generated content. In *CIKM*, 2010.

[8] K. Wang, C. Thrasher, and B. Hsu. Web scale nlp: A case study on url word breaking. In *WWW*, 2011.

[9] K. Wang, C. Thrasher, E. Viegas, X. Li, and B. Hsu. An overview of microsoft web n-gram corpus and applications. In *NAACL*, 2010.

[10] J. Yang, R. Cai, Y. Wang, J. Zhu, L. Zhang, and W. Ma. Incorporating site-level knowledge to extract structured data from web forums. In *WWW*, 2009.

[11] Y. Zhai and B. Liu. Web data extraction based on partial tree alignment. In *WWW*, 2005.

[12] J. Zhu, Z. Nie, J. Wen, B. Zhang, and W. Ma. Simultaneous record detection and attribute labeling in web data extraction. In *SIGKDD*, 2006.