Extracting Researcher Metadata with Labeled Features

Sujatha Das Gollapalli[*] Yanjun Qi[†] Prasenjit Mitra[‡] C. Lee Giles[§]

Abstract

Professional homepages of researchers contain metadata that provides crucial evidence in several digital library tasks such as academic network extraction, record linkage and expertise search. Due to inherent diversity in values for certain metadata fields (e.g., affiliation) supervised algorithms require a large number of labeled examples for accurately identifying values for these fields. We address this issue with *feature labeling*, a recent semi-supervised machine learning technique.

We apply feature labeling to researcher metadata extraction from homepages by combining a small set of expert-provided feature distributions with few fully-labeled examples. We study two types of labeled features: (1) Dictionary features provide unigram hints related to specific metadata fields, whereas, (2) Proximity features capture the layout information between metadata fields on a homepage in a second stage. We experimentally show that this two-stage approach along with labeled features provides significant improvements in the tagging performance. In one experiment with only ten labeled homepages and 22 expert-specified labeled features, we obtained a 45% relative increase in the F1 value for the affiliation field, while the overall F1 improves by 9%.

Keywords: metadata extraction, feature labeling, conditional random fields

1 Introduction

Researcher homepages (also referred to as academic homepages or simply homepages in this paper) form an important resource for information discovery and for obtaining, updating and tracking document collections in digital libraries. Academic homepages typically summarize research and academic interests of researchers and contain other metadata used in tasks such as expertise search, academic network extraction and name disambiguation [1, 16]. Consequently, retrieval of such homepages and extraction of information from them has been of interest, particularly in context of the academic web [3]. In this paper, we address the task of metadata extraction from homepages. That is, given a researcher homepage, our goal is to identify values for a number of pre-defined metadata fields: *employment position*, *university* and *department affiliations* and *contact information* such as email, phone and fax.

The homepage metadata extraction problem can be converted to a sequence labeling (also known as tagging or annotation) problem in a straightforward manner: Given the stream of tokens corresponding to the content on a homepage (We consider textual content and whitespace tokenization), assign to each token a tag/label from the set: { AFFL, EMAIL, FAX, PHN, POS, UNIV, O } where these labels correspond to “affiliation”, “email id”, “fax number”, “phone number”, “employment position”, “university” and “other” fields respectively. An example is illustrated in Table 1.

Although semi-supervised approaches were not investigated previously, metadata extraction from academic homepages was studied before using supervised machine learning [16, 17]. These studies showed that tagging or sequence labeling approaches that capture dependencies among tags out-perform classification-style approaches. This is not surprising since researchers tend to observe certain conventions while placing metadata on their homepages. For instance, it is common to find phone and fax information close together on a researcher homepage. Similarly, employment position information is typically followed by the affiliation information (e.g., “professor” in the “Computer Science department” at “Stanford”). giving rise to dependencies among POS and AFFL tags.

We highlight some challenges in tagging metadata fields on homepages compared to common Natural Language Processing (NLP) tasks such as parts-of-speech tagging that involve tagging fields in general English text [14]:

1. Presence of cue words does not always indicate the metadata of the person that the homepage is about. Although ‘student’ and ‘professor’ are commonly seen values for the *position* field, in Table 1, only ‘student’ needs to be annotated with the ‘POS’ label since this value corresponds to the “owner” of the homepage.

2. A related challenge pertains to similar words occurring with multiple labels where certain labels are more common than the others. For instance, in the above snippet, the first ‘State’ corresponds to a *university* field

[*] The Pennsylvania State University, PA, gsdas@cse.psu.edu. Part of this work was done at NEC Laboratories America.

[†] University of Virginia, VA, yanjun@virginia.edu

[‡] The Pennsylvania State University, PA, pmitra@ist.psu.edu

[§] The Pennsylvania State University, PA, giles@ist.psu.edu

I	**am**	**a**	**student**	**at**	**Penn**	**State**	**and**	**work**	**with**
O	O	O	POS	O	UNIV	UNIV	O	O	O
Professor	**Xxxxx**	**Yyyyy**	**on**	**designing**	**finite**	**state**	**automata**	. . .	
O	O	O	O	O	O	O	O		

Table 1: Homepage Tagging Example

whereas the second 'state' refers to a research problem the student is working on (to be marked as "other"). Since webpages give rise to lengthy sequences of tokens with most of them being "other", we found in our experiments that discriminative terms such as "research, department, student" occur more often with the "other" tag. Learning algorithms that typically use co-occurrence counts may not be able to model such parameters accurately.

3. Values for certain metadata fields exhibit diverse patterns with cue words appearing in various forms and positions (e.g., affiliation values *Department of Computer Science*, *EECS Department*, and *Computer Science and Electrical Engineering Dept*). In addition, patterns with certain cue words may occur rarely in a dataset. For instance, in our dataset, we found that affiliation values containing the term "department" occur about 30% of the time whereas values that contain the term "centre" only occur 1.3% of the time.

How can we account for imbalance in token-label pairs and rare patterns without having to label more examples? For instance, given that we know that the term, "centre" corresponds to affiliation, can we use this information to guide the training process? More generally, *can we extract and incorporate problem-specific hints while training annotation models?* We use *feature labeling*, a recent advancement in semi-supervised learning to answer these questions [8].

Contributions: We study the use of "labeled features" for annotating metadata on researcher homepages. We capture term and layout hints associated with metadata fields via dictionary and proximity labeled features respectively. These hints enable us to train annotation models with fewer training instances. Our contributions are summarized below:

1. First, we propose and evaluate a set of basic features for annotating homepages. In contrast with previous works that use rule-based patterns, noun phrases and visual information, our set of features is minimalistic with domain information separated to dictionary features alone [15, 17].

2. To the best of our knowledge, annotation of researcher homepages using semi-supervised models was not studied before. We adopt the recently proposed feature labeling approach where supervision is provided using (feature, label) distributions which are incorporated into the training process via *posterior regularization*. Our experiments demonstrate the effectiveness of this approach when the number of annotated instances are few.

3. Finally, we study strategies to extract labeled features when labeled training instances are available. In absence of a large number of labeled instances, we show that automatic methods may not be capable of extracting labeled features whose value is comparable to that of expert-specified labeled features in terms of learning better tagging models.

In the next section, we summarize the work closely related to our contributions. In Sections 3 and 4, we describe our methods. Section 5 covers our experimental setup, results and observations while Section 6 concludes our paper.

2 Related work

Information extraction problems are of great interest in the web and natural language processing communities [2, 12, 14]. In particular, metadata extraction from academic homepages was studied for the ArnetMiner project[1] [15] and for CiteSeer[2] [17]. Tang, et al. designed several sets of noun-phrase, dictionary, pattern and term features for identifying the metadata fields. Zheng, et al. instead classify the HTML DOM nodes that correspond to metadata fields using visual features such as font-style and position of the block in the page after which a second stage inter-field probability model is used for the final extraction.

Based on comparisons and observations from these previous studies, we chose Conditional Random Fields (CRFs) for our annotation task. Linear-chain CRFs that address information extraction as sequence tagging problems where models can be trained discriminatively using arbitrary features are shown to be widely successful on various IE tasks [14]. Our focus is on using simpler features and semi-supervised learning with CRFs for homepage annotation.

A recent advance in machine learning pertains to the use "labeled features" for training models [13, 6, 8]. Druck, et al. and Mann, et al. proposed the Generalized Expectation (GE) criterion for using labeled features within discriminative classifiers and taggers [7, 4]. Ganchev, et al. proposed "Posterior Regularization", (PR) a more general framework

[1] http://arnetminer.org/

[2] http://citeseerx.ist.psu.edu

for incorporating "side information" into models for structured prediction by imposing linear constraints on posterior expectations [5]. We use the CRF, labeled features and the PR framework implemented in Mallet, the information extraction package provided by UMass[3].

While the use of labeled features is also referred to as "semi-supervised learning", this is more due to the use of supervision with labeled features as opposed to labeled instances. Semi-supervised learning approaches where supervision is provided at the instance-level is not discussed in this paper. Instead, our focus is on capturing homepage-specific aspects as labeled features for use within the PR framework. Feature extraction for semi-supervised models was previously studied for classification [4, 11] and tagging [6, 7]. However, these works focus on term-based features that frequently correspond to labels. In addition to term features, we design "proximity" features that capture the layout of metadata fields on a homepage.

Our proximity features are similar in spirit to self-labeled features previously studied for tagging problems [10]. Qi, et al. proposed an iterative scheme, where feature vectors in each iteration are augmented with the predicted word-level class label distributions from the previous iteration in a semi-supervised manner. Similarly, we use predictions from a first-stage CRF for use as "labeled features" in a second CRF, effectively combining the two ideas.

3 Methods

3.1 Motivation for a two-stage process

It is reasonable to assume that researchers do not arrange their metadata on their professional homepages arbitrarily. For example, it is unlikely that the phone contact information appears at the top of the page while the fax information appears towards the end. Similarly, it is common to find employment information of a researcher closely listed with the affiliation information (e.g. "I am an assistant professor in the Computer Science department at Stanford"). Indeed, researchers follow certain conventions in placing their metadata and this aspect was captured partially via visual dependencies [17] and transition features [16] in previous research. However, the proposed visual layout features are very intricate while the transition features are limited to a single step in linear-CRFs [14].

In initial experiments, we also noticed that values for certain fields such as phone numbers and fax numbers are often easier to extract than values pertaining to fields like affiliation. Based on these intuitions and observations, we ask the following question:*can the knowledge of certain fields aid in the identification of the other fields*? We seek to answer this question via a two-step approach as follows:

token	S1 preds	S2 features
I	O	nws1POS
am	O	nws1POS
student	POS	s1POS, nws1UNIV
in	O	pws1POS, nws1UNIV
Penn	UNIV	s1UNIV, pws1POS
State	UNIV	s1UNIV
working	O	pws1UNIV
with	O	pws1UNIV
professor	O	O
...		

Table 2: Example demonstrating features added for stage 2 based on stage 1 predicted tags and window size=3 (pw: previous window, nw: next window, s1: stage 1)

1. Use the basic set of features (Table 4) to train a tagger for the first stage.
2. Next, use predicted tags from the first stage tagger as additional features to train a second-stage tagger.

We posit that this two-stage process is better in modeling next labels in addition to previous labels as well as label information within a window rather than just the previous step label dependencies (as in the case of linear-chain CRFs). More precisely, in the second stage, for every token position, we add the closest tag within a window of positions with respect to the current token position. An example is shown in Table 2.

3.2 Stage 1: Training the first CRF tagger

We train a homepage tagger using features corresponding to simple surface patterns, terms and dictionaries. In contrast with previous work that used intricate regular experession patterns and visual features, we chose simple unigram, bigram features based on terms, surface patterns and dictionaries available for this task. We use the following features:

1. **Canonical term features**: These features refer to basic terms corresponding to the textual content on a homepage. We use whitespace tokenization and convert all tokens to lowercase after removing punctuation.
2. **Dictionary features**: We use boolean features corresponding to the presence in field-specific dictionaries. These dictionaries were obtained from previous work related to ArnetMiner where homepage annotation was studied using CRFs and SVMs [16]. These dictionaries comprise a total of 147 cue words often seen with metadata fields. For example, values for the *phone* field usually appear as numeric strings following the cue words, 'phone' or 'ph' (sample words in Table 3).
3. **Surface-form features**: Surface patterns provide valu-

[3] http://mallet.cs.umass.edu/

AFFL	center, centre, college, department, dept, dipartimento, laboratory
UNIV	universiteit, universitat, university, univ
PHN	cell, ext, extn, homephone, mobile, numbers, ph, phonefax, phone
FAX	ext, extn, facsimile, fax, faxno, faxnumber, telefax, tel/fax
EMAIL	contact, email, firstname, lastname, gmail, mail, mailbox, mailto
pre POS	administrative, affiliate, assistant, associate, asst, co, chief, deputy,
POS	president, prof, professor, gradstudent, researcher, scholar, scientist,

Table 3: Sample cue words for different fields

able hints in annotation tasks. For instance, phone numbers are typically numeric fields and values for affiliation fields are often capitalized. We use boolean features indicating if the token matches one of the surface patterns: singleLetter, allletters-capitalized, is-a-capitalized-word, all-digits-in-word, and word-has-digits.

4. **Name-based features**: To capture the empirical observation that most metadata fields on a homepage appear in close proximity with the researcher name, we indicate the presence of a researcher name within a neighborhood of five lines within the line containing the token via this feature.

5. **Sentence delimiter features**: We add sentence boundary features indicating whether the token starts, is inside or ends a sentence. These features are designed to capture the observation that labels do not often extend across line boundaries.

Given a string of terms corresponding to the content of a homepage, let F, G represent feature-types described above. We use subscripts to denote the feature corresponding to a particular position in the text. The feature templates used for training the initial classifier are listed in Table 4. We refer to a CRF model trained on these features as “Basic CRF” or “Stage 1 or S1” in Section 5.

Unigram features	$F_i, i = \{-2, \dots 2\}$
Bigram features	$F_{-1}F_0$ and F_0F_1
Skip features	$F_{-1}F_1$
Conjunction features	$F_{-i}G_0$ and F_0G_i $i = \{-1, 0, 1\}$

Table 4: Feature templates for Stage-1

3.3 Stage 2: Training the second CRF tagger

In our initial experiments with basic features, we found that the tagger was able to accurately identify fields corresponding to phone, fax and e-mail but was not very accurate on fields such as affiliation, position and university (Section 5.3). The second-stage tagger is designed to avail the “most likely correct” fields identified in stage 1 to zero in on other fields in the next round. We obtain the set of predicted tags from the Stage 1 CRF to form the second-set of features for each token position on a homepage.

Table 2 shows an example for generating additional features using predicted tags for a window of size 3. In this table, “pw” and “nw” indicate respectively, the previous and next positions within the specified window size, where the stage 1 tagger labeled a metadata field. For example, “pws1UNIV” indicates that within the previous 3 positions, the stage 1 tagger marked a token with the “UNIV” tag.

The set of features for stage 2 are generated using the conjunction feature templates by combining the predicted tags from stage 1 with term, dictionary and surface pattern features. For example, the features added in stage 2 for the token “Penn” in the example from Table 2 are: `penn_s1UNIV, capitalized_s1UNIV` and `nodict_s1UNIV`.

4 Feature labeling to improve tagging

4.1 Motivation

We further study techniques to improve the tagging performance on the *affiliation*, *position*, and *university* fields. An error analysis of results obtained using the taggers from the previous section indicated two reasons for low numbers on these fields in our datasets:

1. Cue words that are indicative of metadata fields occur with the “other” tag more often than with the specific field potentially making it hard for the model to estimate weights for the associated parameters accurately. For instance, the term, “student” occurs with the “POS” tag only 23% of the time in our dataset, whereas the term “university” occurs with the “UNIV” tag about 36% of the time. The remaining percent corresponds to their occurrence with the “O” tag.

2. The affiliation, position and university fields exhibit various patterns in their values with a severe imbalance in the training examples for each pattern. For example, there are about 300 labeled instances having affiliation field values with the term “department”, whereas only 14 labeled instances have the term “centre” in their affiliation values.

Supervised learning algorithms require a number of labeled instances to estimate accurately the weights corresponding to feature parameters. For taggers, edge transition parameters need to be additionally estimated. Compared to classification tasks where labels are assigned at an instance level (e.g., for a document or an image), labeling examples for sequence tagging where each token position in the sequence needs to be marked up involves considerably more human effort. However, recent research indicated that while labeling instances is hard, labeling features is considerably easier and faster for an expert in the domain [11].

Given the advancement in semi-supervised learning with labeled features, we ask the question: *What kind of hints can we provide via labeled features to the learning algorithms to enable better tagging of researcher metadata on a homepage?*

4.2 Background

Labeled features were studied by Mann, Druck and McCallum for both classification and tagging problems [7, 4]. They proposed adding "supervision" to learning algorithms by providing (feature, label) affinities rather than fully-annotated instances. Consider the example in Table 1. Even without annotating the entire snippet, from domain knowledge, one can expect the correct label for the token "student" to be "POS", "most" of the time. This hint can be imposed as a soft preference or a constraint by specifying the (feature, label) distribution. For example, the labeled feature "student POS:0.8, O:0.2" indicates a preference for marking the token "student" with the label "POS" 80% of the time. Generalized Expectation (GE) and Posterior Regularization (PR) are two frameworks studied previously for imposing such preferences in discriminative models [8, 5]. We choose the PR framework since it handles more general constraints and was found to be better performing in our experiments. In the next subsection, we briefly describe posterior regularization with labeled features for completeness.

4.2.1 Posterior Regularization

Using the notation from [14], let the pair $(\mathbf{x}, \mathbf{y})$ represent an instance for sequence labeling where $\mathbf{x}$ corresponds to the token sequence and $\mathbf{y}$ represents the label sequence for $\mathbf{x}$. The feature functions in CRFs take the form: $f_k(y_t, y_{t-1}, \mathbf{x}, t)$. If $\theta = \{\lambda_k\}$ represents the parameter vector corresponding to $k = 1 \dots K$ features, for a linear-chain CRF the conditional distribution is given by:

$$p_\theta(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_k \lambda_k F_k(\mathbf{x}, \mathbf{y})\right) \tag{4.1}$$

where $F_k(\mathbf{x}, \mathbf{y}) = \sum_t f_k(\mathbf{x}, y_t, y_{t-1}, t)$ and the partition function $Z(\mathbf{x}) = \sum_{\mathbf{y}} \exp\left(\sum_k \lambda_k F_k(\mathbf{x}, \mathbf{y})\right)$.

The regularized, conditional log likelihood function optimized in CRFs is given by:

$$l(\theta) = \sum_{i=1}^{N} \log p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) - \sum_{k=1}^{K} \frac{\theta_k^2}{2\sigma^2} \tag{4.2}$$

In the above equation, Euclidean norm is used for regularizing the parameter vector, θ with the regularization factor given by $\frac{1}{2\sigma^2}$. In posterior regularization framework, data-dependent constraints are encoded as model posteriors on the observed data. Using the Markov assumption, the feature expectations in CRF can be written as $\sum_{\mathbf{y}} p_\theta(\mathbf{y}|\mathbf{x}) F_k(\mathbf{x}, \mathbf{y}) = \sum_t \sum_{y_t, y_{t-1}} p(y_t, y_{t-1}|\mathbf{x}) f_k(y_t, y_{t-1}, \mathbf{x}, t)$.

$$\sum_{\mathbf{y}} p_\theta(\mathbf{y}|\mathbf{x}) F_k(\mathbf{x}, \mathbf{y}) =$$

$$\sum_t \sum_{y_t, y_{t-1}} p(y_t, y_{t-1}|\mathbf{x}) f_k(y_t, y_{t-1}, \mathbf{x}, t)$$

The distributions specified via labeled features are converted into expectation constraints in the PR framework as follows: Let $\phi(x_v, \mathbf{y})$ represent the value of the feature expectation estimated by the model when $x = v$. If b represents the target expectation, the constraints corresponding to x can be written as

$$Q_{x_v} = \{q_{x_v}(\mathbf{y}) : \mathbf{E}_q[\phi(x_v, \mathbf{y})] \leq b\}$$

Let $\mathbf{X}, \mathbf{Y}$ represent the training instances and Q, the desired distribution space representing all constraints, the objective for PR captures the KL-divergence between Q and the model posteriors to be minimized as:

$$J_Q(\theta) = l(\theta) - \mathrm{KL}(Q||p_\theta(\mathbf{Y}|\mathbf{X})) \tag{4.3}$$

The posterior expectations are specified as linear constraints during the parameter estimation process in the PR framework. More details on PR, optimization issues and other forms of the objective function are described in [5].

4.3 Labeled features for homepages

We study two types of labeled features:

1. **Dictionary features** capture terms that commonly occur with certain fields. For example, (student, *position*), (dept, *affiliation*), etc.

2. **Proximity features** capture the layout conventions on researchers homepages. For example, using the notation described in Section 3.1, (department_pws1POS, *affiliation*) indicates a preference for *affiliation* if the current token is "department" and previous tokens within a window were marked as *position*.

In the description above, we specified the majority labels for the features. More generally, labeled features refer to label-probability distributions. For example, "(student *position*:0.9, *other*:0.1)" indicates a distribution on labels, *position* and *other*. Mann and McCallum showed that given limited annotation time, models that were trained using expert-specified labeled features out-perform other semi-supervised approaches that use fully-labeled instances in their experiments [4]. Sample expert-designed features pertaining to *affiliation* are shown in Table 5.

Dictionary features	**Proximity features**
laboratory	nws1UNIV
centre	capitalized_pws1POS
college	capitalized_nws1UNIV
department	daffl_pws1POS
dipartimento	capitalized_pws1UNIV
institute	science_nws1UNIV

Table 5: Expert-specified dictionary and proximity features for affiliation values. s1$*$ refers to the predicted label obtained from the first-stage CRF. For example, "daffl_pws1POS" reads as the token is found in the affiliation dictionary and has a predicted tag, POS from stage-1 within the previous window

4.4 Extracting labeled features automatically

A labeled feature is a specification of a feature along with a probability distribution related to labels associated with it. *Is it possible to extract labeled features automatically given labeled data?* This question was briefly addressed in context of classification by using mutual information between the features and labels and Latent Dirichlet Allocation [4]. In context of tagging, frequently occuring features with a given label "that do not also occur frequently with other labels" were used to extract features automatically [6, 7].

To obtain label-probability distributions automatically, a few options were studied in the same works: (1) Majority distribution (**MAJ**) where the majority label associated with a feature gets the majority of probability mass whereas the remaining mass is distributed uniformly among the remaining labels; (2) Schapire distribution (**SCH**) where the majority of the mass is uniformly distributed among all labels associated with a feature while the remaining mass is uniformly distributed among the non-associated labels [13]; and (3) Feature-voted (**VOTE**) distributions where co-occurrence counts of (feature, label) pairs in the training data are normalized to obtain probability distributions. We also propose and evaluate two variant schemes: (4) (**MAJ***) is a variation of **MAJ** where only associated labels in the training data are taken into account. That is, among the associated labels, the winning label gets the majority of the probability mass, whereas the remaining mass is distributed uniformly among the remaining associated labels, and (5) (**MAJ****) is similar to MAJ* except that the winning label gets the majority of the mass only if it is a clear winner (has co-occurrence with the feature more than or equal to twice that of other associated labels).

As an example, let $(A, 10)$, $(B, 6)$, $(C, 0)$ $(D, 0)$ be the co-occurrence counts in the training data for a feature for which a distribution is to be generated for labels A, B, C and D. If we choose majority probability mass value to be 0.9, the label distributions obtained with the different schemes are given by:

- MAJ: {A:0.9 B:0.033 C:0.033, D:0.033}
- SCH: {A:0.45 B:0.45 C:0.05, D:0.05}
- VOTE: {A:0.625 B:0.375}
- MAJ*: {A:0.9 B:0.1}
- MAJ**: {A:0.5 B:0.5 }

Using the (feature, label) co-occurrences in training data we study feature extraction with (1) Information Gain (**IG**), and (2) Frequency (**TF**). We found that Mutual Information (**MI**) selected rare words for each label, such as univerisity

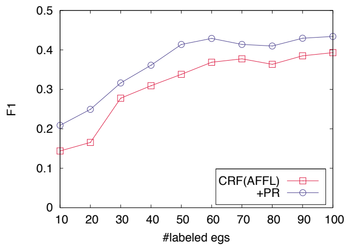

Figure 1: Performance on the affiliation field with different numbers of initial training examples.

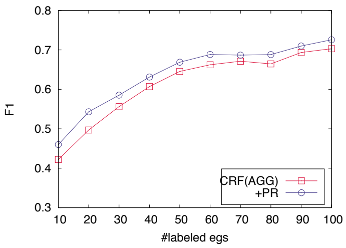

Figure 2: Aggregate performance with different numbers of initial training examples.

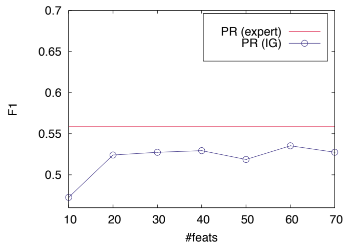

Figure 3: PR with varying number of IG-based dictionary features (#expert-features=22).

names among the top words and did not benefit from PR. This "known" issue with MI and details on IG and MI can be found in [9]. Sample dictionary features chosen using these strategies based on the instances in our dataset (Section 5) are shown in Table 6. In our experiments section, we compare the features extracted using different extraction methods and probability distribution options with the expert-designed labeled features.

Rank	**TF**	**MI**	**IG**
1	university	bogazici	professor
2	professor	fudan	university
3	department	patras	associate
4	computer	sofia	assistant
5	associate	ogi	department
6	assistant	bozen	director
7	science	bolzano	computer
8	research	hokkaido	science
9	engineering	tung	student
10	student	albany	researcher
26	phd	marie	graduate
27	graduate	kth	phd
28	center	harbor	center
29	california	potsdam	electrical
30	scientist	clara	member

Table 6: Sample terms based on frequency (TF), mutual information (MI) and information gain (IG) are shown in this table.

5 Experiments

5.1 Datasets and settings

We summarize our experiments on the ArnetMiner profile tagging dataset.This dataset comprises of 898 annotated researcher pages collected by Tang, et al. for studying record linkage across homepages and DBLP records [15, 16]. To the best of our knowledge, this is the only **publicly-available** dataset available for our problem[4]. All experiments were performed on a 16-core, 800MHz, 32GB RAM, AMD Opteron, Linux server with default parameter settings in Mallet (Section 2). First, the stage 1 CRF is trained with the basic set of features described in Section 3.2. Next, predictions from stage 1 are used to generate additional features for the stage 2 CRF. Predictions on the test data are made in a transductive setting with PR. That is, the PR framework treats the test instances as unlabeled instances and imposes the labeled feature constraints during training. The model obtained after this phase is used to obtain predictions for the test instances.

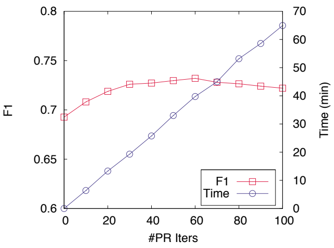

Figure 5: PR (dictionary constraints) with different number of training iterations (Time in minutes vs. F1).

The dictionary labeled features can be used with PR for both the stage 1 and stage 2 CRFs whereas by design, the proximity features can be used only in the second stage. We train the CRF until convergence is obtained on the training split of the data in each experiment. For about 600 training instances, the CRF training time was 30 minutes on average for both stage 1 and stage 2 CRFs. An EM-style optimization algorithm is used in the posterior regularization framework [5]. We analyzed the performance versus running time trade-off for a few sample runs (Figure 5) and set the number of iterations for the PR part to 50 in the rest of the experiments. The window size was set to 10 for stage 2 experiments since the expert-labeled features were generated based on this size.

As is common in annotation evaluation [9], we use the F1 measure that provides a single metric capturing both precision and recall to compare models. In all experiments we show the aggregate performance over all fields (entries marked by 'AGG'). For some experiments, we highlight performance on the affiliation field since we found it to be the most difficult to extract among all the fields (entries marked by 'AFFL').

5.2 Expert-labeled features

The *affiliation* and *university* field-specific dictionary terms obtained from ArnetMiner (Section 3.2) were directly used as dictionary labeled features. For proximity features, we manually examined the homepages in the dataset to derive a list of common layout conventions. For example, "affiliation fields are often preceded by position and followed by university information", or "phone information is usually listed before fax information". Next, dictionary terms and surface patterns were used to form conjunctions within a window with the label set $T = \{AFFL, POS, UNIV\}$ (e.g., capitalized_pwPOS, department_nwPHN). This list was manually examined to obtain a subset of features that satisfy the layout conventions. These "expert" lists of 22 dictionary and 20 proximity features along with their label distributions are available upon request (Table 5 contains a sample).

5.3 Results and observations

Figures 1 and 2 illustrate the benefits of using expert-

[4] Datasets and code are available upon request.

Field	Precision		Recall		F1	
	Basic	Best	Basic	Best	Basic	Best
AFFL	0.6670	0.4571	0.4302	0.7095	0.5219	**0.5554** (+6.4%)
EMAIL	0.9178	0.8889	0.8136	0.8693	0.8624	0.8788 (+1.9%)
FAX	0.9543	0.9501	0.9295	0.9406	0.9417	0.9453
PHN	0.9370	0.9310	0.8899	0.9296	0.9128	0.9303 (+1.9%)
POS	0.8048	0.7470	0.5995	0.6835	0.6870	**0.7138** (+3.9%)
UNIV	0.7203	0.6596	0.5827	0.7336	0.6432	**0.6940** (+7.9%)

Table 7: F1 values (three-fold cross-validation) for basic and the "best" performing set of features. The best performance was obtained by imposing all constraints in stage 2 using predictions from stage 1 with dictionary PR.

specified dictionary labeled features to provide "supervision" when the number of labeled homepage instances to train the initial CRF is small. In this situation, as the plots indicate, (feature, label) distributions are able to effectively boost the tagging performance. The figures show improvements in affiliation F1 from 0.1438 to 0.2088 and in aggregate F1 from 0.4222 to 0.4601 when only ten labeled examples are available. As the number of labeled examples increase, the initial CRF becomes more accurate, with a reduction in the boost with PR.

We summarize our three-fold cross-validation experiments comparing Stage 1 and Stage 2 CRF in Table 9. About 600 labeled examples are available in each run and the overall boost in F1 is not as high as in the previous experiment. However, notice the field-specific improvements obtained by enforcing the expert-specified feature label constraints using PR in Figure 4. Both dictionary and proximity features provide improvements over that obtained using CRF alone for both the stages. A comparison of the F1 values between the "S1" and "S2" rows of Table 9 validates our intuition regarding the use of two stages for annotating homepages (Section 3.1). It appears that performance-wise a two-staged approach along with proximity constraints is comparable to using a single stage process with dictionary constraints and combination schemes do not provide large enhancements. The field-specific improvements comparing the basic stage 1 CRF and our best-performing model are shown in Table 7. The entries where the improvement in F1 is more than 2% are marked in bold in this table. Using a paired t-test, the improvements in $F1$ values are statistically significant for p-value$= 0.05$. Imposing constraints via labeled features increases recall for all the fields at the cost of a dip in precision, with the F1 improving for all the fields.

Setting	AFFL	Agg	Agg/O
Basic CRF	0.5219	0.7937	0.7615
Posterior Regularization			
Expert	0.5548	0.8124	0.7835
IG (VOTE)	0.5493	0.8059	0.7757
IG (MAJ)	0.5224	0.7944	0.7623
IG (SCH)	0.5512	0.8062	0.7760
IG (MAJ*)	0.5230	0.7944	0.7623
IG (MAJ**)	0.5423	0.8046	0.7742
TF (VOTE)	0.5415	0.8030	0.7723
TF (MAJ)	0.5173	0.7903	0.7576
TF (SCH)	0.5514	0.8061	0.7759
TF (MAJ*)	0.5261	0.7966	0.7649
TF (MAJ**)	0.5459	0.8048	0.7745

Table 8: Three-fold cross-validation F1 values for top-30 dictionary labeled features (#expert features=22).

Our experiments with the feature extraction and distribution assignment schemes described in Section 4 are summarized in Table 8 for dictionary labeled features. Using information gain along with feature-voted distribution seems

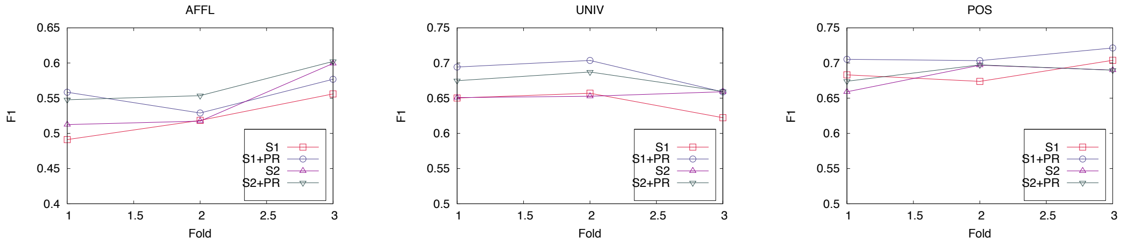

Figure 4: F1 variation across folds for *affiliation*, *university* and *position* fields.

to be the best performing scheme among all the configurations although this is lower than that obtained with expert-specified features. Increasing the number of IG features does not help as can be observed in Figure 3. It is likely that IG features might get more accurate when a large amounts of labeled data is available for estimating them. However, the boost obtained using labeled features via the PR framework seems to reduce as more labeled instances become available (Figure 1).

Setting	**AFFL**	**Agg**	**Agg/O**
S1	0.5219	0.7937	0.7615
S1+PR (Dictionary)	0.5548	0.8124	0.7835
Predictions from Stage 1 (basic)			
S2	0.5431	0.7974	0.7656
S2+PR (Proximity)	0.5679	0.8047	0.7743
S2+PR (All)	0.5527	0.8065	0.7767
Predictions from Stage 1 (basic + dict PR)			
S2	0.5303	0.8001	0.7689
S2+PR (Proximity)	0.5621	0.8082	0.7784
S2+PR (All)	0.5554	0.8147	0.7862

Table 9: Three-fold cross-validation F1 values for Stage 1 (S1) and Stage 2 (S2) models. Expert-specified labeled features were used.

6 Conclusions

We studied *feature labeling* for annotating metadata on researcher homepages. We proposed dictionary-based features to capture field-specific hints whereas proximity features capture the layout information among metadata fields. Our proximity features are different from labeled features typically studied in previous work in that they are designed to be used in a second-stage model using predictions from a first-stage model. We showed that posterior regularization can effectively impose the dictionary and proximity hints during the training process to obtain significant improvements in tagging performance when the labeled examples available are limited.

To the best of our knowledge, we are the first to investigate feature labeling for its application for metadata extraction on webpages as opposed to NLP tasks studied before. We complement term features with problem-specific, layout-based labeled features using predictions from a first-stage tagger. In addition, we showed experimentally that unlike NLP tasks investigated previously, a large amount of tagged data might be required for extracting labeled features using automatic methods that match the performance obtained with expert-specified labeled features. However, this requirement invalidates the benefit of labeled features since supervised methods can be directly used when large amount of tagged data is available.

Acknowledgments We gratefully acknowledge partial support from the National Science Foundation.

References

[1] Krisztian Balog, T. Bogers, L. Azzopardi, M. de Rijke, and A. van den Bosch. Broad expertise retrieval in sparse data environments. In *SIGIR*, 2007.

[2] Chia-Hui Chang, Mohammed Kayed, Moheb Ramzy Girgis, and Khaled F. Shaalan. A survey of web information extraction systems. *TKDE*, October 2006.

[3] Sujatha Das, Cornelia Caragea, Prasenjit Mitra, and C. Lee Giles. Improving academic homepage classification using unlabeled data. In *WWW*, 2013.

[4] Gregory Druck, Gideon Mann, and Andrew McCallum. Learning from labeled features using generalized expectation criteria. In *SIGIR*, 2008.

[5] Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *JMLR*, 2010.

[6] Aria Haghighi and Dan Klein. Prototype-driven learning for sequence models. In *HLT-NAACL '06*, 2006.

[7] Gideon S. Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *ACL*, 2008.

[8] Gideon S. Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *JMLR*, 2010.

[9] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

[10] Yanjun Qi, Pavel P. Kuksa, Ronan Collobert, Kunihiko Sadamasa, Koray Kavukcuoglu, and Jason Weston. Semi-supervised sequence labeling with self-learned features. In *ICDM*, 2009.

[11] Hema Raghavan, Omid Madani, and Rosie Jones. Active learning with feedback on features and instances. *JMLR*, 2006.

[12] Sunita Sarawagi. Information extraction. *Foundations and Trends in Databases*, March 2008.

[13] R. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting, 2002.

[14] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. November 2010.

[15] Jie Tang, Duo Zhang, and Limin Yao. Social network extraction of academic researchers. *ICDM*, 2007.

[16] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *KDD*, 2008.

[17] Shuyi Zheng, Ding Zhou, Jia Li, and C. Lee Giles. Extracting author meta-data from web using visual features. *ICDM Workshops*, 2007.