# Minimally-Supervised Extraction of Entities from Text Advertisements

**Sameer Singh**
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003
sameer@cs.umass.edu

**Dustin Hillard**
Advertising Sciences
Yahoo! Labs Silicon Valley
Santa Clara, CA 95054
dhillard@yahoo-inc.com

**Chris Leggetter**
Advertising Sciences
Yahoo! Labs Silicon Valley
Santa Clara, CA 95054
cjl@yahoo-inc.com

## Abstract

Extraction of entities from ad creatives is an important problem that can benefit many computational advertising tasks. Supervised and semi-supervised solutions rely on labeled data which is expensive, time consuming, and difficult to procure for ad creatives. A small set of manually derived constraints on feature expectations over unlabeled data can be used to *partially* and *probabilistically* label large amounts of data. Utilizing recent work in constraint-based semi-supervised learning, this paper injects light weight supervision specified as these "constraints" into a semi-Markov conditional random field model of entity extraction in ad creatives. Relying solely on the constraints, the model is trained on a set of unlabeled ads using an online learning algorithm. We demonstrate significant accuracy improvements on a manually labeled test set as compared to a baseline dictionary approach. We also achieve accuracy that approaches a fully supervised classifier.

## 1 Introduction

Growth and competition in web search in recent years has created an increasing need for improvements in organic and sponsored search. While foundational approaches still focus on matching the exact words of a search to potential results, there is emerging need to better understand the underlying *intent* in queries and documents. The implicit intent is particularly important when little text is available, such as for user queries and advertiser creatives.

This work specifically explores the extraction of named-entities, i.e. discovering and labeling phrases in ad creatives. For example, for an ad "Move to San Francisco!", we would like to extract the entity *san francisco* and label it a CITY. Similarly, for an ad "Find DVD players at Amazon", we would extract *dvd players* as a PRODUCT and *amazon* as a ORGNAME. The named-entities provide important features to downstream tasks about what words and phrases are important, as well as information on the intent. Much recent research has focused on extracting useful information from text advertisement creatives that can be used for better retrieval and ranking of ads. Semantic annotation of queries and ad creatives allows for more powerful retrieval models. Structured representations of semantics, like the one studied in our task, can be directly framed as information extraction tasks, such as segmentation and named-entity recognition.

Information extraction methods commonly rely on labeled data for training the models. The human labeling of ad creatives would have to provide the complete segmentation and entity labels for the ads, which the information extraction algorithm would then rely on as the truth. For entity extraction from advertisements this involves familiarity with a large number of different domains, such as electronics, transportation, apparel, lodging, sports, dining, services, *etc*. This leads to an arduous and time consuming labeling process that can result in noisy and error-prone data. The problem is further compounded by the inherent ambiguity of the task, leading to the human editors often presenting conflicting and incorrect labeling.

Similar problems, to a certain degree, are also faced by a number of other machine learning tasks where completely relying on the labeled data leads to unsatisfactory results. To counter the noisy and sparse labels, *semi-supervised learning* meth-



*Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 73–81,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

ods utilize unlabeled data to improve the model (see (Chapelle et al., 2006) for an overview). Furthermore, recent work on *constraint-based semi-supervised learning* allows domain experts to easily provide additional light supervision, enabling the learning algorithm to learn using the prior domain knowledge, labeled and unlabeled data (Chang et al., 2007; Mann and McCallum, 2008; Bellare et al., 2009; Singh et al., 2010).

Prior domain knowledge, if it can be easily expressed and incorporated into the learning algorithm, can often be a high-quality and cheap substitute for labeled data. For example, previous work has often used dictionaries or *lexicons* (lists of phrases of a particular label) to bootstrap the model (Agichtein and Ganti, 2004; Canisius and Sporleder, 2007), leading to a *partial* labeling of the data. Domain knowledge can also be more *probabilistic* in nature, representing the probability of certain token taking on a certain label. For most tasks, labeled data is a convenient representation of the domain knowledge, but for complex domains such as structured information extraction from ads, these alternative easily expressible representations may be as effective as labeled data.

Our approach to solving the the named entity extraction problem for ads relies completely on domain knowledge not expressed as labeled data, an approach that is termed *minimally supervised*. Each ad creative is represented as a semi-Markov conditional random field that probabilistically represents the segmentation and labeling of the creative. External domain knowledge is expressed as a set of targets for the expectations of a small subset of the features of the model. We use *alternating projections* (Bellare et al., 2009) to train our model using this knowledge, relying on the rest of the features of the model to "dissipate" the knowledge. Topic model and co-occurrence based features help this propagation by generalizing the supervision to a large number of similar ads.

This method is applied to a large dataset of text advertisements sampled from a variety of different domains. The minimally supervised model performs significantly better than a model that incorporates the domain knowledge as hard constraints. Our model also performs competitively when compared to a supervised model trained on labeled data from a similar domain (web search queries).

Background material on semi-CRFs and constraint based semi-supervised learning is summarized in Section 2. In Section 3, we describe the problem of named entity recognition in ad creatives as a semi-CRF, and describe the features in Section 4. The constraints that we use to inject supervision into our model are listed in Section 5. We demonstrate the success of our approach in Section 6. This work is compared with related literature in Section 7.

## 2 Background

This section covers introductory material on the probabilistic representation of our model (semi-Markov conditional random fields) and the constraint-driven semi-supervised method that we use to inject supervision into the model.

### 2.1 Semi-Markov Conditional Random Fields

Conditional Random Fields (CRFs) (Lafferty et al., 2001) use a Markov random field to model the conditional probability $P(\mathbf{y}|\mathbf{x})$. CRFs are commonly used to learn sequential models, where the Markov field is a linear-chain, and $\mathbf{y}$ is a linear sequence of labels and each label $y_i \in \mathcal{Y}$. Let $\mathbf{f}$ be a vector of *local feature functions* $\mathbf{f} = \langle f^1, \ldots, f^K \rangle$, each of which maps a pair $(\mathbf{x}, \mathbf{y})$ and an index $i$ to a measurement $f^k(i, \mathbf{x}, \mathbf{y}) \in \Re$. Let $\mathbf{f}(i, \mathbf{x}, \mathbf{y})$ be the vector of these measurements, and let $\mathbf{F}(\mathbf{x}, \mathbf{y}) = \sum_i^{|\mathbf{x}|} \mathbf{f}(i, \mathbf{x}, \mathbf{y})$. CRFs use these feature functions in conjunction with the parameters $\theta$ to represent the conditional probability as follows:

$$P(\mathbf{y}|\mathbf{x}, \theta) = \frac{1}{Z(\mathbf{x})} e^{\theta \cdot \mathbf{F}(\mathbf{x}, \mathbf{y})}$$

where $Z(x) = \sum_{\mathbf{y}'} e^{\theta \cdot \mathbf{F}(\mathbf{x}, \mathbf{y}')}$.

For sequential models where the same labels appear within a sequence as contiguous blocks (*e.g.*, named entity recognition) it is more convenient to represent these blocks directly as *segments*. This representation was formulated as *semi-Markov conditional random fields (Semi-CRFs)* in (Sarawagi and Cohen, 2004). The *segmentation* of a sequence is represented by $\mathbf{s} = \langle s_1, \ldots, s_p \rangle$ where each *segment* $s_j = \langle t_j, u_j, y_j \rangle$ consists of a *start position* $t_j$, an *end position* $u_j$, and a *label* $y_j \in \mathcal{Y}$. Similar to the CRF, let $\mathbf{g}$ be the vector of *segment feature*

*functions* $\mathbf{g} = \langle g^1, \ldots, g^K \rangle$, each of which maps the pair $(\mathbf{x}, \mathbf{s})$ and an index $j$ to a measurement $g^k(j, \mathbf{x}, \mathbf{s}) \in \Re$, and $\mathbf{G}(\mathbf{x}, \mathbf{s}) = \sum_j^{|\mathbf{s}|} \mathbf{g}(j, \mathbf{x}, \mathbf{s})$. The conditional probability is represented as:

$$P(\mathbf{s}|\mathbf{x}, \theta) = \frac{1}{Z(\mathbf{x})} e^{\theta \cdot \mathbf{G}(\mathbf{x}, \mathbf{s})}$$

where $Z(x) = \sum_{\mathbf{s}'} e^{\theta \cdot \mathbf{G}(\mathbf{x}, \mathbf{s}')}$. To assert the Markovian assumption, each $g^k(j, \mathbf{x}, \mathbf{s})$ only computes features based on $\mathbf{x}$, $s_j$, and $y_{j-1}$[1].

An exact inference algorithm was described in (Sarawagi and Cohen, 2004), and was later improved to be more efficient (Sarawagi, 2006).

### 2.2 Constraint Driven Learning Using Alternating Projections

Recent work in semi-supervised learning uses constraints as external supervision (Chang et al., 2007; Mann and McCallum, 2008; Bellare et al., 2009; Singh et al., 2010). These external constraints are specified as constraints on the expectations of a set of auxiliary features $\mathbf{g}' = \{g'_1, \ldots, g'_k\}$ over the unlabeled data. In particular, given the targets $\mathbf{u} = \{u_1, \ldots, u_k\}$ corresponding to the auxiliary features $\mathbf{g}'$, the constraints can take different forms, for example $\mathbb{L}_2$ penalty ($\frac{1}{2\beta}\|u_i - \sum_j E_p[g'_i(x_j, s)]\|_2^2 = 0$), $\mathbb{L}_1$ box constraints ($|u_i - \sum_j E_p[g'_i(x_j, s)]| \leq \beta$) and Affine constraints[2] ($E_p[g'_i(x, s)] \leq u_i$). In this work, we only use the affine form of the constraints.

For an example, using domain knowledge, we may know that token "arizona" should get the label STATE in at least half of the occurrences in our data. To capture this, we introduce an auxiliary feature $g'$ : [[`Label`=STATE *given* `Token`="arizona"]]. The affine constraint is written as $E_p[g'(x, y)] \geq 0.5$.

These constraints have been incorporated into learning using Alternating Projections (Bellare et al., 2009). Instead of directly optimizing an objective function that includes the constraints, this method considers two distributions, $p_\lambda$ and $q_{\lambda,\mu}$, where $p_\lambda(\mathbf{s}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} e^{\lambda \cdot \mathbf{G}(\mathbf{x}, \mathbf{s})}$ is the usual semi-Markov model, and $q_{\lambda,\mu} = \frac{1}{Z(\mathbf{x})} e^{(\lambda \cdot \mathbf{G}(\mathbf{x}, \mathbf{s}) + \mu \cdot \mathbf{G}'(\mathbf{x}, \mathbf{s}))}$ is an auxiliary distribution that satisfies the constraints and has low divergence with the model $p_\lambda$.

[1] i.e. $g^k(j, \mathbf{x}, \mathbf{s})$ can be written as $g^k(y_{j-1}, \mathbf{x}, s_j)$

[2] where $E_p[g]$ represents the expectation of $g$ over the unlabeled data using the model $p$.

In the batch setting, parameters $\lambda$ and $\mu$ are learned using an EM-like algorithm, where $\mu$ is fixed while optimizing $\lambda$ and vice versa. Each of the updates in these steps decomposes according to the instances, leading to a stochastic gradient based online algorithm, as follows:

1. For $t = 1, \ldots, T$, let $\eta = \frac{1}{t+t_0}$ where $t_0 = 1/\eta_0$, $\eta_0$ the initial learning rate. Let labeled and unlabeled data set sizes be $m$ and $n - m$ respectively. Let the initial parameters be $\lambda^0$ and $\mu^0$, and $\alpha$ be the weight of $\mathbb{L}_2$ regularization on $\lambda$.

2. For a new *labeled* instance $x_t$ with segmentation $s_t$, set $\mu^t = \mu^{t-1}$ and $\lambda^t = \lambda^{t-1} + \eta \left[ \mathbf{g}(x_t, s_t) - E_{p_{\lambda^{t-1}}}[\mathbf{g}(x_t, s)] - \frac{\alpha \lambda^{t-1}}{n} \right]$.

3. For a new *unlabeled* instance $x_t$, $\mu^t = \mu^{t-1} + \eta \left[ \frac{u}{(n-m)} - E_{q_{\lambda^{t-1}, \mu^{t-1}}}[\mathbf{g}'(x_t, s)] \right]$ and $\lambda^t = \lambda^{t-1} + \eta \left[ E_{q_{\lambda^{t-1}, \mu^{t-1}}}[\mathbf{g}(x_t, s)] - E_{p_{\lambda^{t-1}}}[\mathbf{g}(x_t, s)] - \frac{\alpha \lambda^{t-1}}{n} \right]$.

Online training enables scaling the approach to large data sets, as is the case with ads. In our approach we rely only on unlabeled data ($m = 0$, and step 2 of the above algorithm does not apply).

## 3 Model

Most text ads consist of a brief *title* and an accompanying *abstract* that provides additional information. The objective of our paper is to extract the named-entity phrases within these titles and abstracts, then label them with a *type* from a predetermined taxonomy. An example of such an extraction is shown in Fig 1.

We represent the ad creatives as a sequence of individual tokens, with a special token inserted between the title and the abstract of the ad. The distribution over possible phrases and labels of the ad is expressed as a semi-Markov conditional random field, as described earlier in Section 2.1.

### 3.1 Label Taxonomy

In most applications of CRFs and semi-CRFs, the domain of labels is a fixed set $\mathcal{Y}$, where each label indexes into one value. Instead, in our approach, we represent our set of labels as a taxonomy (tree). The labels higher in the taxonomy are more generic (for

**Ad Title:** Bradley International Airport Hotel

**Ad Abstract:** Marriott Hartford, CT Airport hotel - free shuttle service & parking.

**Output:**

Bradley International Airport Hotel

Marriott Hartford, CT Airport hotel free shuttle service & parking.

| Label | Segment |
|---|---|
| PLACE: AIRPORT | Bradley International |
| BUSINESS: TRAVEL | Hotel |
| ORGNAME: LODGING | Marriott |
| PLACE: CITY | Hartford |
| PLACE: STATE | CT |
| BUSINESS: TRAVEL | hotel |
| PRODUCT: TRAVEL | shuttle service & parking. |

Figure 1: **Example Prediction:** An example of an ad creative (title and abstract), along with a set of probable extracted entities. Note that even in this relatively simple example, there is some ambiguity about what is the correct segmentation and labeling.

instance, PLACE) and the labels lower in the taxonomy are more specific (for instance, STATE may be a child of PLACE). The taxonomy of labels that we use for tagging phrases is shown in Figure 2.

When the model predicts a label for a segment, it can be from any of the levels in the tree. The benefits of this is multi-fold. First, this allows the model to be flexible in predicting labels at a lower (or higher) level based on its confidence. For example, the model may have enough evidence to label "san francisco" a CITY, however, for "georgia" it may not have enough context to discriminate between STATE or COUNTRY, but could confidently label it a PLACE. Secondly, this also allows us to design the features over multiple levels of label granularity, which leads to a more expressive model. Expectation constraints can be specified over this expanded set of features, at any level of the taxonomy.

In order to incorporate the nested labels into our model, we observe that every feature that fires for a non-leaf label should also fire for all descendants of that label, *e.g.* every feature that is active for label PLACE should also be active for a label CITY, COUNTRY, *etc* [3]. Following the observation, for every feature $g^k(\mathbf{x}, \langle t_j, u_j, y_j \rangle)$ that is active, we also fire $\forall y' \in \text{desc}(y_j), g^k(\mathbf{x}, \langle t_j, u_j, y' \rangle)$[4]. The same procedure is applied to the constraints.

## 4 Features

Our learning algorithm relies on constraints $\mathbf{g}'$ as supervision to extract entities, but even though constraints are designed to be generic they do not cover the whole dataset. The learning algorithm needs to propagate the supervision to instances where the constraints are not applicable, guided by the set of feature functions $\mathbf{g}$. More expressive and relevant features will provide better propagation. Even though these feature functions represent the "unsupervised" part of the model (in that they are only dependent on the unlabeled sequences), they play an important role in propagating the supervision throughout the dataset.

### 4.1 Sequence and Segment Features

Our first set of features are the commonly used features employed in linear-chain sequence models such as CRFs and HMMs. These consist of factors between each token and its corresponding label, and neighboring labels. They also include transition factors between the labels. These are *local feature functions* that are defined only over pairs of token-wise

[3] Note that this argument works similarly for the taxonomy represented as a DAG, where the descendants are of a node are all nodes reachable from it. We do not explore this structure of the taxonomy in this paper.

[4] This example describes when $g^k(y_{j-1}, \mathbf{x}, s_j)$ ignores $y_{j-1}$. For the usual case $g^k(y_{j-1}, \mathbf{x}, s_j)$, features between all pairs of descendants of $y_{j-1}$ and $y_j$ are enabled.

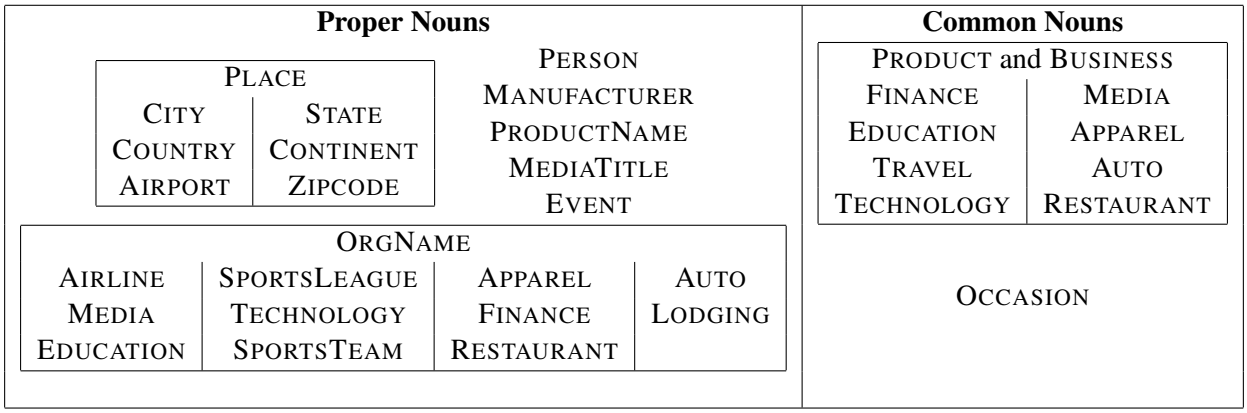


Figure 2: **Label Taxonomy:** The set of labels that are used are shown grouped by the parent label. PRODUCT and BUSINESS labels have been merged for brevity, i.e. there are two labels of each child label shown (e.g. PRODUCT: AUTO and BUSINESS: AUTO). An additional label OTHER is used for the tokens that do not belong to any entities.

labels $y_j$ and $y_{j-1}$. To utilize the semi-Markov representation that allows features over the predicted segmentation, we add the segment length and prefix/suffix tokens of the segment as features.

### 4.2 Segment Clusters

Although the sequence and segment features capture a lot of useful information, they are not sufficient for propagation. For example, if we have a constraint about the token "london" being a CITY, but not about "boston", the model can only rely on similar contexts between 'london" and 'boston" to propagate the information. To allow more complicated propagation to occur, we use features based on a clustering of segments.

The segment cluster features are based on similarity between segments from English sentences. A large corpus of English documents were taken from web, from which 5.1 billion unique sentences were extracted. Using the co-occurrence of segments in the sentences as a distance measure, K-Means is used to identify clusters of segments as described in (Pantel et al., 2009). The cluster identity of each segment is added as a feature to the model, capturing the intuition that segments that appear in the same cluster should get the same label.

### 4.3 Topic Model

Most of the ads lie in separate domains with very little overlap, for example travel and electronics. Additional information about the domain can be very useful for identifying entities in the ad. For example, consider the token "amazon". It may be difficult to discern whether the token refers to the geographical region or the website from just the features in the model, however given that the domain of the ad is travel (or conversely, electronics), the choice becomes easier.

The problem of domain identification is often posed as a document classification task, which requires labeled data to train and thus is not applicable for our task. Additionally, we are not concerned with accurately specifying the exact domain of each ad, instead any information about similarity between ads according to their domains is helpful. This kind of representation can be obtained in an unsupervised fashion by using topic cluster models (Steyvers and Griffiths, 2007; Blei et al., 2003). Given a large set of unlabeled documents, topic models define a distribution of topics over each document, such that documents that are similar to each other have similar topic distributions.

The LDA (Blei et al., 2003) implementation of topic models in the Mallet toolkit (McCallum, 2002) was used to construct a model with 1000 topics for a dataset containing 3 million ads. For each ad, the discrete distribution over the topics, in conjunction with each possible label, was added as a feature. This captures a potential for each label given an approximation of the ad's domain captured as topics.

## 5 Constraints

Constraints are used to inject light supervision into the learning algorithm and are defined as targets $u$ for expectations of features $G'$ over the data. Any feature that can be included in the model can be used as a constraint. This allows us to capture a variety of different forms of domain knowledge, some of which we shall explore in this section.

Labeled data $x_l, s_l$ can be incorporated as a special case when constraints have a target expectation of 1.0 for the features that are defined only for the sequence $x_l$ and with segmentation $s_l$. This allows us to easily use labeled data in form of constraints, but in this work we do not include any labeled data. A more interesting case is that of partial labeling, where the domain expert may have prior knowledge about the probability that certain tokens and/or contexts result in a specific label. These constraints can cover more instances than labeled data, however they only provide partial and stochastic labels. All of the constraints described in this section are also included as simple features.

Many different methods have been suggested in recent work for finding the correct target values for the feature expectations. First, if ample labeled data is available, features expectations can be calculated, and assumptions can be made that the same expectations hold for the unlabeled data. This method cannot be applied to our work due to lack of labeled data. Second, for certain constraints, the prior knowledge can be used directly to specify these values. Third, if the constraints are an output of a previous machine learning model, we can use that model's confidence in the prediction as the target expectation of the constraint. Finally, a search for the ideal values of the target expectations can be performed by evaluating on small evaluation data. Our target values for feature expectations were set based on domain knowledge, then adjusted manually based on minimal manual examination of examples on a small held-out data set.

### 5.1 Dictionary-Based

Dictionary constraints are the form of constraints that apply to the feature between an individual token and its label. For a set of tokens in the dictionary, the constraints specify which label they are likely to be. Dictionaries can be easily constructed using various sources, for example product databases, lexicons, manual collections, or predictions from other models. These dictionary constraints are often used to bootstrap models (Agichtein and Ganti, 2004; Canisius and Sporleder, 2007) and have also been used in the ads domain (Li et al., 2009). For our application, we rely on dictionary constraints from two sources.

First, the predictions of a previous model are used to construct a dictionary. A model for entity extraction is trained on a large amount of labeled search query data. The domain and style of web queries differs from advertisements, but the set of labels is essentially the same. The supervised query entity extraction model is used to infer segments and labels for the ads domain, and each of the predicted segments are added to the dictionary of the corresponding predicted label. Even though the predictions of the model are not perfect (see Section 6.1) the predictions of some of the labels are of high precision, and thus can be used for supervision in form of noisy dictionary constraints.

The second source of prior information for dictionary constraints are external databases. Lists of various types of places can be obtained easily, for example CITY, COUNTRY, STATE, AIRPORT, etc. Additionally, product databases available internally to our research group are used for MANUFACTURERS, BRANDS, PRODUCTS, MEDIATITLE, etc. Some of these databases are noisy, and the constraints based on them are given lower target expectations.

### 5.2 Pattern-Based

Prior knowledge can often be easily expressed as patterns that appear for a specific domain. Pattern based matching has been used to express supervision for information extraction tasks (Califf and Mooney, 1999; Muslea, 1999). The usual use case involves a domain expert specifying a number of "prototypical" patterns, while additional patterns are discovered based on these initial patterns.

We incorporate noisy forms of patterns as constraints. Simple regular expression based patterns were used to identify and label segments for a few domains (*e.g.* "flights to {PLACE}" and "looking for {PRODUCT}?"). We do not employ a pattern-discovery algorithm for finding other contexts; the model propagates these labels, as before, using the

features of the rest of the model. However if the output of a pattern-discovery algorithm is available, it can be directly incorporated into the model as additional constraints.

### 5.3 Domain-Based

A number of label-independent constraints are also added to avoid unrealistic segmentation predictions. For example, an expectation over segment lengths was included, which denotes that the segment length is usually 1 or 2, and almost never more than 6. A constraint is also added to avoid segments that overlap the separator token between title and abstract by ensuring that the segment that includes the separator token is always of length 1 and of label OTHER. Finally, an additional constraint ensures that the label OTHER is the most common label.

## 6 Results

The feature expectations of the model are calculated with modifications to an open source semi-CRF package[5]. We collect two datasets of ad creatives randomly sampled from Yahoo!'s ads database: a smaller dataset contains 14k ads and a larger dataset of 42k ads. The ads were not restricted to any particular domain (such as travel, electronics, etc.). The average length of the complete ad text was $\sim$14 tokens. Preprocessing of the text involved lower-casing, basic cleaning, and stemming.

The training time for each iteration through the data was $\sim$90 minutes for the smaller dataset and $\sim$360 minutes for the larger dataset. Inference over the dataset, using Viterbi decoding for semi-CRFs, took a total of $\sim$8 and $\sim$32 minutes. The initial learning rate $\eta$ is set to $10.0$.

### 6.1 Discussion

We compare our approach to a baseline "Dictionary" system that deterministically selects a label based on the dictionaries described in Section 5.1. A segment is given a label corresponding to the dictionary it appears in, or OTHER if it does not appear in any dictionary. In addition, we compare to an external supervised system that has been trained on tens-of-thousands of manually-annotated search queries that use the same taxonomy (the same system as used in Section 5.1 to derive dictionaries). This CRF-based model contains mostly the same features as our unsupervised system, and approximates what a fully supervised system might achieve, although it is trained on search queries. Results for our approach and these two systems are presented in Table 1. Our evaluation data consists of 2,157 randomly sampled ads that were manually labeled by professional editors. This labeled data size was too small to sufficiently train a supervised semi-CRF model that out-performed the dictionary baseline for our task (which consists of 45 potential labels).

We measure the token-wise accuracy and macro F-score over the manually labeled dataset. Typically, these metrics measure only exact matches between the true and the predicted label, but this leads to cases where the model may predict PLACE for a true CITY. To allow a "partial credit" for these cases, we introduce "weighted" version of these measures, where a predicted label is given $0.5$ credit if the true label is its direct child or parent, and $0.25$ credit if the true label is a sibling. Our F-score measures the recall of all true labels except OTHER and similarly the precision of all predicted labels except OTHER. We focus on these labels because the OTHER label is mostly uninformative for downstream tasks. The token-wise accuracy over all labels (including OTHER) is included as "Overall Accuracy".

Our method significantly outperforms the baseline dictionary method while approaching the results obtained with the sophisticated supervised model. Overall accuracy is 50% greater than the dictionary baseline, and comes within 10% of the supervised model[6]. Increasing unlabeled data from 14k to 42k ads provides an increase in overall accuracy and non-OTHER precision, but somewhat reduces recall for the remaining labels. We also include the F2-score which gives more weight to recall, because we are interested in extracting informative labels for downstream models (which may be able to compensate for a lower precision in label prediction). Our model trained on 14k samples out-performs the query-based supervised model in terms of F2, which is promising for future work that will incorporate predicted labels in ad retrieval and ranking systems.

[5] Available on http://crf.sourceforge.net/

[6] Comparisons and trends for normal and weighted measures are consistent throughout the results.

Table 1: **Evaluation:** Token-wise accuracy and F-score for the methods evaluated on labeled data (Normal / *Weighted*)

| Metric | Dictionary | Our Method (14k) | Our Method (42k) | Query-based Sup. Model |
|---|---|---|---|---|
| Overall Accuracy | 0.454 / *0.466* | 0.596 / *0.627* | 0.629 / *0.649* | 0.665 / *0.685* |
| non-OTHER Recall | 0.170 / *0.205* | 0.329 / *0.412* | 0.271 / *0.325* | 0.286 / *0.342* |
| non-OTHER Precision | 0.136 / *0.163* | 0.265 / *0.333* | 0.297 / *0.357* | 0.392 / *0.469* |
| F1-score | 0.151 / *0.182* | 0.293 / *0.368* | 0.283 / *0.340* | 0.331 / *0.395* |
| F2-score | 0.162 / *0.195* | 0.313 / *0.393* | 0.276 / *0.331* | 0.303 / *0.361* |

## 7 Related Work

Extraction of structured information from text is of interest to a large number of communities. However, in the ads domain, the task has usually been simplified to that of classification or ranking. Previous work has focused on retrieval (Raghavan and Iyer, 2008), user click prediction (Shaparenko et al., 2009; Richardson et al., 2007; Ciaramita et al., 2008), ad relevance (Hillard et al., 2010) and bounce rate prediction (Sculley et al., 2009). As far we know, our method is the only one that aims to solve a much more complex task of segmentation and entity extraction from ad creatives. Supervised methods are a poor choice to solve this task as they require large amounts of labeled ads, which is expensive, time-consuming and noisy. Most semi-supervised methods also rely on *some* labeled data, and scale badly with the size of unlabeled data, which is intractable for most ad databases.

Considerable research has been undertaken to exploit forms of domain knowledge other than labeled data to efficiently train a model while utilizing the unlabeled data. These include methods that express domain knowledge as constraints on features, which have shown to provide high accuracy on natural language datasets (Chang et al., 2007; Chang et al., 2008; Mann and McCallum, 2008; Bellare et al., 2009; Singh et al., 2010). We use the method of alternating projections for constraint-driven learning (Bellare et al., 2009) since it specifies constraints on feature expectations instead of less intuitive constraints on feature parameters (as in (Chang et al., 2008)). Additionally, the alternating projection method is computationally more efficient than Generalized Expectation (Mann and McCallum, 2008) and can be applied in an online fashion using stochastic gradient.

Our approach is most similar to (Li et al., 2009), which uses semi-supervised learning for CRFs to extract structured information from user queries. They also use a constraint-driven method that utilizes an external data source. Their method, however, relies on labeled data for part of the supervision while our method uses only unlabeled data. Also, evaluation was only shown for a small domain of user queries, while our work does not restrict itself to any specific domain of ads for evaluation.

## 8 Conclusions

Although important for a number of tasks in sponsored search, extraction of structured information from text advertisements is not a well-studied problem. The difficulty of the problem lies in the expensive, time-consuming and error-prone labeling process. In this work, the aim was to explore machine learning methods that do not use labeled data, relying instead on light supervision specified as constraints on feature expectations. The results clearly show this *minimally-supervised* method performs significantly better than a dictionary based baseline. Our method also approaches the performance of a supervised model trained to extract entities from web search queries. These findings strongly suggest that domain knowledge expressed in forms other than directly labeled data may be preferable in domains for which labeling data is unsuitable.

The most important limitation lies in the fact that specifying the target expectations of constraints is an ad-hoc process, and robustness of the semi-supervised learning method to noise in these target values needs to be investigated. Further research will also explore using the extracted entities from advertisements to improve downstream sponsored search tasks.

## References


Eugene Agichtein and Venkatesh Ganti. 2004. Mining reference tables for automatic text segmentation. In *KDD: ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 20–29, New York, NY, USA.

Kedar Bellare, Gregory Druck, and Andrew McCallum. 2009. Alternating projections for learning with expectation constraints. In *UAI: Conference on Uncertainty in Artificial Intelligence*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal on Machine Learning Research*, 3:993–1022.

Mary Elaine Califf and Raymond J. Mooney. 1999. Relational learning of pattern-match rules for information extraction. In *AAAI / IAAI '99: National conference on Artificial intelligence and the Innovative Applications of Artificial Intelligence conference*, pages 328–334.

Sander Canisius and Caroline Sporleder. 2007. Bootstrapping information extraction from field books. In *EMNLP-CoNLL: Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 827–836.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *ACL: Annual meeting of the Association for Computational Linguistics*, pages 280–287.

Ming-Wei Chang, Lev Ratinov, Nicholas Rizzolo, and Dan Roth. 2008. Learning and inference with constraints. In *AAAI: National Conference on Artificial Intelligence*, pages 1513–1518.

O. Chapelle, B. Schölkopf, and A. Zien, editors. 2006. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. The MIT Press, September.

Massimiliano Ciaramita, Vanessa Murdock, and Vassilis Plachouras. 2008. Online learning from click data for sponsored search. In *WWW: International World Wide Web Conference*.

Dustin Hillard, Stefan Schroedl, Eren Manavoglu, Hema Raghavan, and Chris Leggetter. 2010. Improving ad relevance in sponsored search. In *WSDM: International conference on Web search and data mining*, pages 361–370.

John Lafferty, Andrew Mccallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML: International Conference on Machine Learning*, pages 282–289.

Xiao Li, Ye-Yi Wang, and Alex Acero. 2009. Extracting structured information from user queries with semi-supervised conditional random fields. In *SIGIR: International Conference on research and development in information retrieval*, pages 572–579. ACM.

Gideon S. Mann and Andrew McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *ACL: Annual meeting of the Association for Computational Linguistics*, pages 870–878.

Andrew McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Ion Muslea. 1999. Extraction patterns for information extraction tasks: A survey. In *AAAI: Workshop on Machine Learning for Information Extraction*, pages 1–6.

Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *EMNLP: Conference on Empirical Methods in Natural Language Processing*, pages 938–947.

Hema Raghavan and Rukmini Iyer. 2008. Evaluating vector-space and probabilistic models for query to ad matching. In *SIGIR Workshop on Information Retrieval in Advertising (IRA)*.

Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *WWW: International World Wide Web Conference*.

Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *NIPS: Neural Information Processing Systems*.

Sunita Sarawagi. 2006. Efficient inference on sequence segmentation models. In *ICML: International Conference on Machine Learning*, pages 793–800.

D. Sculley, Robert G. Malkin, Sugato Basu, and Roberto J. Bayardo. 2009. Predicting bounce rates in sponsored search advertisements. In *KDD: ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 1325–1334.

Benyah Shaparenko, Ozgur Cetin, and Rukmini Iyer. 2009. Data driven text features for sponsored search click prediction. In *AdKDD: Workshop on Data mining and audience intelligence for advertising*.

Sameer Singh, Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Constraint-driven rank-based learning for information extraction. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*.

Mark Steyvers and Tom Griffiths. 2007. *Probabilistic Topic Models*. Lawrence Erlbaum Associates.