Extraction of (Key,Value) Pairs from Unstructured Ads

Sunandan Chakraborty[1] and **Lakshminarayanan Subramanian**[2] and **Yaw Nyarko**[3]
[12]Department of Computer Science
[3]Department of Economics
New York University
New York, USA
[123]Center for Technology and Economic Development (CTED)
NYU Abu Dhabi, UAE
{sunandan,lakshmi}@cs.nyu.edu and yaw.nyarko@nyu.edu

Abstract

In this paper, we focus on the problem of extracting structured labeled data from short unstructured ad-postings from online sources like Craigslist, where ads are posted on various topics, such as job postings, rentals, car sales etc. A fundamental challenge in addressing this problem is that most ad-postings are highly unstructured, short-text postings written in an informal manner with no inherent grammar or well-defined dictionary. In this paper, we propose unsupervised and supervised algorithms for extracting structured data from unstructured ads in the form of (key, value) pairs where the *keys* naturally represent topic-specific features in the ads. The unsupervised algorithm is centered around building an affinity graph, using the words from a topic-specific corpus of such ads where the edge weights represent affinities between words; the (key, value) extraction algorithm identifies specific groups of words in the affinity graph corresponding to different classes of key attributes. The supervised algorithm uses a Conditional Random Field based training algorithm to identify specific structured (key, value) pairs based on pre-defined topic-specific structural data representations of ads. Based on a corpus of car and apartment ad-postings from Craigslist, the unsupervised algorithm reported an accuracy of 67.74% and 68.74% for car and apartment ads respectively. The supervised algorithm demonstrated an improved performance with accuracies of 74.07% and 72.59% respectively.

Introduction

This paper aims to address the problem of extracting structured information from unstructured ad-postings in the form of *(key,value)* pairs where a *key* represents a specific attribute about the underlying ad-posting with a corresponding *value*. This problem is largely relevant in the content of the current Web where large volumes of user-defined information in online portals like Craigslist are in the form of unstructured short-message ad-postings. These ad-postings are written in an informal style without a well defined dictionary or grammar and the quality of the textual data tends to be highly variable and noisy; in addition, different users may use different abbreviations and formats for conveying the same information. While humans can easily interpret such ad-postings, it is hard for an automated tool to perform data analysis on them. To enhance machine analysis of such unstructured postings, a critical problem to address is to be able to convert these ad-postings into structured data with defined keys and values.

Given a corpus of ads under a topic (such as car ads from Craigslist), our goal is to extract a structured feature-based representation for all the ad-postings. Specifically, we want to convert any unstructured ad to a set of (key, value) pairs where *key* refers to a specific feature corresponding to the topic and *value* represents specific descriptive information for that feature. Solving this problem has several important practical ramifications including enabling a range of advanced search options which can be tailored for topic-specific features; currently many of the advanced search options for unstructured ads are constrained only for specific features (such as cost [price, rent etc.], model, year) with simple field extractors. Our approach can enable the users to search using more specific terms and constraints as well as improve the quality of the results based on existing options.

Our problem is different in spirit from prior work on extracting structured information from unstructured text (Grenager, Klein, and Manning 2005)(Haghighi and Klein 2006)(Michelson and Knoblock 2008)(Druck, Mann, and McCallum 2008) due to our focus on unstructured ads and the task of extracting (key,value) pairs from these postings. While specific prior work (Grenager, Klein, and Manning 2005)(Michelson and Knoblock 2008) have also examined unstructured ad-postings, the underlying focus of these works have been different from our work. The focus of (Grenager, Klein, and Manning 2005) was to extract field structures from unstructured texts using small amounts of prior knowledge while(Michelson and Knoblock 2008) proposed building a relational database from such texts using external knowledge bases.

This paper proposes an unsupervised and a supervised algorithm for (key, value) extraction from unstructured ads. In most ads the object advertised is described using some standard features of the object. For example, for apartment ads such features include apartment size, apartment rent, location, number of bedrooms etc. Every ad under a specific topic contains such a generic template and a particular ad is a full description of those features but presented without any proper format. We try to capture this inherent structure in the form of (key,value) pairs, where *keys* are the features (e.g. apartment rent) and *values* are a specific value (e.g. $2000) as advertised. The converted structured form of an ad is represented as a set of various (key,value) pairs. The unsupervised algorithm constructs a word affinity graph, where the edge weights represent the affinities between words as measured by the mutual information metric between word pairs. We define three specific classes of keys in the unsupervised algorithm: binary keys, numeric keys and descriptive keys. Binary keys represent specific features where the value is a binary output on whether the feature is present or not. Numeric features involve keys where the value represents a numeric output. Descriptive keys are ones where the key represents a broad category with a possible set of values (e.g., color of a car). The identification of the keys and the values from this graph are determined by analyzing specific affinity patterns of words in the graph with their neighbors.

The supervised approach is trained on a manually annotated training set, where we explicitly assume that the set of keys for a given topic is known previously. We implement a Conditional Random Field (CRF) based method to annotate the ad terms with descriptive keys. This supervised model computes the best sequence of keys – from a predetermined set of topic-specific keys which can best describe the ads – given the word sequence of an ad. Although, a supervised approach is costlier and difficult to generalize over variety of topics, it demonstrated better performance.

We applied the unsupervised approach on a corpus of 12,984 ads on cars and 10,784 apartment ads downloaded from Craigslist. Evaluating on a manually annotated test set, the unsupervised method achieved an accuracy of 67.74% for cars and 68.74% for apartment ads. The supervised approach was trained on a manually annotated training set of 600 ads from each topic. The supervised algorithm yielded an accuracy of 74.07% for car ads and 72.59% for the apartment ads.

Related Work

The problem of extracting structure from unstructured text has been studied in prior work. The works by Grenager et al (2005), Haghighi and Klein (2006) are focused on field extraction from Craiglist ads. However, their end goals were different from the work presented in this paper. Also, they focused only on apartment ads and their label set was slightly different. Hence, these works are not directly comparable to the work presented here. Michelson and Knoblock (2008) tried to solve a similar problem creating relational data sets from unstructured and ungrammatical posts, like Craigslist and eBay for better search for such texts. They used external reference sets to build this relational dataset. Another related work is by Kim et al (2012), where they present an unsupervised information extraction system for short listings on the web by building a domain-specific semantic model. Wang et al (2011) proposed a novel topic model called Structural Topic Model, which they evaluated on a corpus of Craigslist ads on apartment rental (Grenager, Klein, and Manning 2005). Sailer et al (2008) addressed a similar problem where they tried to identify structural pattern in call centre data, which is usually unstructured, noisy and heterogeneous, using CRFs. Probst et al (2007) proposed a system which is quite similar to our problem. Their system was targetted at extracting *attribute-value* pairs from ads of sporting goods product. Blei et al (2001) is another example such an work where topic segmentation was carried out in unstructured text. Several prior works have also analyzed the problem of adding structure to unstructured text; these include McCallum (2005), Pradhan et al (2003), Haghighi & Klein(2006), Druck et al (2008) (2009). The problem we address in this paper is different in spirit from these works due to our focus on unstructured ad-postings and the specific task of (key,value) pair extraction.

Another broad class of closely related work is on summarizing natural language text where the goal is to create a summary of a larger text without losing crucial information. One fundamental difference between this class of works and our work is the nature of the output. While we present output in $< label, value >$ format, the output in summarization tasks is usually in natural text form. There has been numerous works proposed to achieve automatic text summarization. One of the major variation of text summarization is *single document summarization* and *multiple document summarization* (McKeown et al. 1999)(Radev, Jing, and Budzikowska 2000)(Evans, Mckeown, and Klavans 2005). There has been different kind of machine learning techniques used to address the issues. For example, Naive-Bayes method (Kupiec, Pedersen, and Chen 1995)(Aone et al. 1999), Neural network (Svore 2007), Hidden Markov model (Conroy and O'leary 2001) etc.

Problem Statement

Given $S = \{S_1, S_2, ..., S_n\}$ where each S_i is an unstructured ad from the Craigslist on a specific topic t. Here, t can be cars, apartments etc. The end goal is to convert each S_i into a set of $\{< key_{ik}, value_{ik} >\}$, where each key_{ik} represents some feature of the ad and $Value_{ik}$ its value. This means that the original posting is approximately represented using K keys and their corresponding values in a tabular structure. An example of such a representation of the ad in Table 1 is shown in Table 3.

We assume that there are three different groups of keys. They can be *descriptive*, *binary* and *numeric*. Descriptive keys are those who can have numerous values, binary keys have only two values, *yes/no* or *present/not present*. Finally, numeric keys have numerical values. Table 1 shows a typical ad from Craigslist. Here, color can be a descriptive key whose value in this case is black. Other possible values are blue, red, silver etc. Power window, AM/FM Radio are examples of binary keys. As they are mentioned in this ad, their values are yes, else it would have been no. For some other possible binary key, say "power seats", the value in this case is "no". Finally, miles, price are examples of numerical key and their values are mentioned close to their occurrences in the ad.

From this example it can also be seen that for descriptive features, the keys sometimes appear in the ad (e.g. Transmission) and on other occasions it does not (e.g. car model

Table 1: A sample ad from Craigslist

Great car for our New England weather, 2004 BMW 325xi, Color Black, 114k Miles, 4 Door, All Wheel Drive, Automatic Transmission, Alloy Wheels, Fog lamps, Sun/Moon Roof, Air Conditioning, Cruise Control, Heated Seats, Leather Seats, Power Door Locks, Power Mirrors, Power Windows, Rear Defrost, AM/FM Radio, CD Player, Keyless Entry, Trip/Mileage Computer, Driver Air Bag, Passenger Air Bag, price $11,488

or make). For binaries, presence of the key determines the value. And, for numerical keys, usually the key and the their values both are present in the ad. Different ads follow different formats. Often the keys with same meaning is expressed using different terms, e.g. **price of the car** can be mentioned using the terms *price, cost, offer* etc.

Table 2: Different keys and their occurrence in the ads

Label	Type	Example
Color	Numeric	Grey with black interior shiney red paint the color is black Red with black racing stripe
Price	Desc	Best Offer $5000 Cost $2952 Asking $2200 firm value is 3000$ price for quick sale 3500 $
Miles	Numeric	Under 62,000 miles approx 170k miles has 144,000 miles
Power steering	Binary	new power steering Power Steering, Cruise ... Control Power Steering - Power ... Brakes

Datasets

To implement and evaluate our methods, we applied them on a corpus of Craigslist ads on 2 topics, cars and apartment rentals. We collected 12,984 cars ads and 10,784 apartment rental ads from Craigslist. We downloaded these ads directly from the Craigslist website [1] belonging to the cities of Boston, New York, Chicago and Los Angeles. We used around 80% of the ads to build the models and kept the rest to test them. The test set was manually labeled to measure the performance.

Unsupervised Method

The unsupervised method proposed in this paper is a graph-based approach. The graph constructed using the words in the ads – as vertices – capture the relationships between the words, i.e. an edge between the vertices shows the affinity between them and thus, the constructed graph is called *Affinity Graph*. This approach is devised on the assumption that across topics, the keys can be classified into 3 different categories: descriptive, binary and numeric. The affinity graph is designed in a way that these different classes of keys can be easily detected from the graph. Hence, we propose a deterministic rule-based inference mechanism to detect the set of topic-specific keys from the affinity graph and their corresponding values.

Table 3: Corresponding $< key, value >$ pairs from Table 1 (partial). Desc: Descriptive, Num: Numeric, Bin: Binary

Label (Type)	Value
Make (Desc)	BMW
Color (Desc)	black
Miles (Num)	114,000
Transmission (Desc)	automatic
Alloy wheels (Bin)	yes
Air conditioning (Bin)	yes
Leather Seats (Bin)	yes
Poor window (Bin)	yes
AM/FM radio (Bin)	yes
CD player (Bin)	yes
Price (Num)	11,488

Affinity Graph

The entire ad corpus can be represented as a set of K unique terms or words $W = w_1, w_2, ..., w_k$. Here, each w_i can a key or a value belonging to one of the classes, descriptive, binary or numeric. Intuitively, a key-value pair, belonging to a particular class will demonstrate a strong relationship between themselves. We design a graphical structure that can capture this inter word relationships. We construct a graph $G_{aff} = (W, E)$, we call it Affinity Graph, where the vertices correspond to each w_i in W and edges are constructed between vertices when the corresponding words in the ad demonstrate a strong relationship. The edge weight between two words w_i and w_j is computed as the mutual information between them defined as,

$$MI(w_i, w_j) = log\frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

where, $p(w_i)$ is the probability of the term w_i occurring in an ad and the joint probability of two terms are the probability of the two terms occurring next to each other (adjacent) in the corpus. Here, w_i are all the words appearing in the entire ad corpus excluding the stop words and very frequently appearing words. Also, all numeric terms in the ads are replaced by a variable vertex w_{NUM}, instead of representing the exact numeric value.

And edges between w_i, w_j are defined as,

$$e(w_i, w_j) \in Efor\{w_i, w_j\} \in W,$$
$$ifMI(w_i, w_j) \gg 0$$

with the edge weight,

[1] www.craigslist.org

$$w_{e(w_i,w_j)} = MI(w_i, w_j)$$

As, mutual information is symmetric, the edges in the affinity graph are undirected. By construction, the Affinity graph will be a collection of disjoint subgraphs.

Identification of Labels-Values Pairs from Affinity Graph

The vertices in the affinity graph are of 3 types: descriptive, binary and numeric, based on the orientation of the terms within the affinity graph. The hypothesis behind this clustering is based on the following claims,

- Descriptive keys will form a star-shaped component in the graph, where the center term is the key and the terms connected to it are its values
- Numeric keys will have a strong relationship with the w_{NUM} vertex. As the values of a numeric key is non standard and not fixed (e.g. price of a car is not a fixed entity, whereas for descriptive keys, like *color* of a car, have fixed values), this approach can only identify the keys. The value for a particular numeric key has to be extracted from individual ads
- Binary keys will form strongly related components (in terms of edge weights) and all the terms in that component will have similar weights. Binary keys do not have separate values. The presence of the binary keys is indicative of their values

Based on these assumptions, we propose an algorithm which takes the affinity graph as input and classifies each vertex into one of the following 3 classes, *descriptive*, *binary* or *numeric*. The proposed algorithm is a deterministic algorithm and is presented as Algorithm 1.

The classification of the vertices into the three classes is based on the degree of the vertices. As per the hypotheses, the different types of keys will have different kinds of orientation in the graph. For example, if a vertex has many edges, the vertex is more likely to be a descriptive key. This makes the association betweeen the words (or an edge between the words) key indicator in identifying the (key,value) pairs. So, the graph needs to be pruned to eliminate the edges which do not demonstrate sufficient affinity betweeen the words. An edge $e(w_i, w_j)$ was pruned if $MI(w_i, w_j) > \lambda_{threshold}$. The $\lambda_{threshold}$ is determined empirically by observing the edge weights of sampled word pairs. Two samples were taken; one with known cases of high affinity and the other with no associations between the pairs. We took the means and the standard deviation of both the distributions. Assuming that both are normally distributed, we took the $\lambda_{threshold}$ as the average of the data point value at the 95^{th} percentile of the weak association distribution and point at the 5^{th} percentile of the strong association distribution. If $qnorm(p, \mu, \sigma)$ represents the quantile function of a normal distribution with mean μ and standard deviation σ then,

$$\lambda_{threshold} = \frac{qnorm(0.05,\mu_s,\sigma_s)+qnorm(0.95,\mu_w,\sigma_w)}{2}$$

where μ_s,μ_w,σ_s,σ_w are the means and the standard deviations of the strong and the weak association edge-weight distributions. Using this threshold edge weight value, we pruned the affinity graph to have more well defined components. The association in the new components is stronger and well-defined, eliminating all the weak associations.

The classification of the words into various *keys* and *values* can be done by detecting the orientation of the new components and is done by computing the conditional probability of a vertex t given its neighbors for all the vertices in a component (line 4-5 in Algorithm 1) . The cumulative score (line 6) for a vertex is the sum of this conditional probability from all its neighbors. If the conditional probabilities $P(w_i|w_j) \approx P(w_j|w_i)$ means that whenever w_i or w_j occurs they occur together. Hence, they together constitute a binary key. This concept can be generalized to include binary keys containing two or more words.

On the other hand for descriptive keys, the key words (e.g. color of a car) should have comparatively large number of neighbors and the corresponding value words (e.g. black, blue etc.) should only be associated with the key word. Hence, if $P(w_i|w_j) \gg P(w_j|w_i)$ then w_j occurs only with w_i but w_i can occur with other terms. This translates into w_i is a descriptive key and w_j is one of the possible value term as occurred in the corpus.

Finally, the numeric keys can be identified if $P(w_i|w_{num}) \approx P(w_{num}|w_i)$, which means that if a term only occurs with a numeric entry in the ads, then that term is a numeric key. Three lists are created $Label_{Desc}, Label_{Bin} and Label_{Num}$, where all the corresponding keys are stored. The function returns these lists at the end.

Performance

The graph-based method was applied on a corpus of 10,000 Craigslist ads on cars and 8,000 ads on apartment rentals to learn a set of keys (descriptive, binary and numeric) for the two different topics. The set of keys learned from the training were applied on a set of 2,984 car and 2,784 apartment rental ads to evaluate the performance. (key,value) pairs from test sets were manually extracted beforehand and used as a golden set. The golden set was created by a human annotator, who manually inspected the test sets and identified all the descriptive, binary and numeric labels. To evaluate the unsupervised approach, the affinity graph constructed during the training phase was used to extract the (key,value) pairs from the testing set. The extracted key-value pairs were compared against the manually crafted golden set and the performance was calculated using precision-recall values. The F-value computed from the precision and recall values for the car and apartment ad sets are shown in Figure 1 and 2 respectively.

This unsupervised approach gave an accuracy of 67.74% for car ads and 68.74% for the apartment ads. The error rate was comparatively lower for numerical and binary keys but it was higher for descriptive keys. The reason behind this low accuracy is mainly due to the fact that often the descriptive key terms are not mentioned within the text. As an example, for car ads the ‘value’ of the ‘key’ ‘color’ is usually

Algorithm 1

```
procedure GetKeyValue
Input: Affinity graph
Output: Set of keys and their values
    $Labels_{Desc} \leftarrow \{\}$
    $Labels_{Bin} \leftarrow \{\}$
    $Labels_{Num} \leftarrow \{\}$
    for each $w_i$ in $W$ do
        $score(w_i) = 0$
    for each $w_i$ in $W$ do
        for each $x$ in $neighbor(w)$ do
            $P(x|w) = \frac{1}{deg(w)}$
            $score(x) = score(x) + P(x|w)$
        for each $e(w_i, w_j)$ in $E$ do
            if $score(w_i) \gg score(w_j)$ then
                $Labels_{Desc}.add(w_i)$
                $Value[w_i] \leftarrow w_j$
            else$score(w_i) \ll score(w_j)$
                $Labels_{Desc}.add(w_j)$
                $Value[w_i] \leftarrow w_i$
            if $score(w_i) \approx score(w_j)$ then
                $Labels_{Bin}.add(w_i)$
                $Labels_{Bin}.add(w_i, w_j)$
            if $score(w_i) \approx score(w_{num})$ then
                $Labels_{Num}.add(w_1)$
    return $Label_{Desc}, Label_{Bin}, Label_{Num}$
```

mentioned, like, black, silver but the actual name of the key (in this case 'color') does not appear in the text. As a result, in many cases descriptive keys are misclassified as binary keys. On the other hand, some binary keys with same words in them are classified as descriptive instead of binary. An example of such an error is 'power window', 'power brakes', 'power steering' etc. Instead of classifying them as separate binary keys, the algorithm classified 'power' as a descriptive key and 'window', 'brakes' and 'steering' as its values. This happened because the word 'power' was in all of them and satisfied the condition of being a descriptive key.

The goal of this work is to build a workable system which can convert unstructured ads into a structured tabular form.

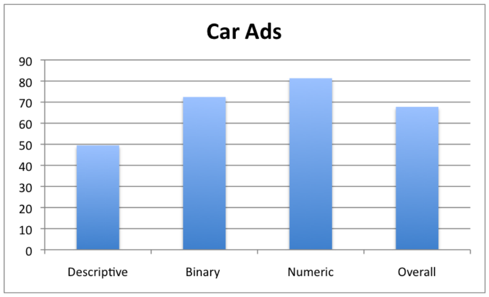

Figure 1: Total accuracy (F-measure) for car ads and under each category

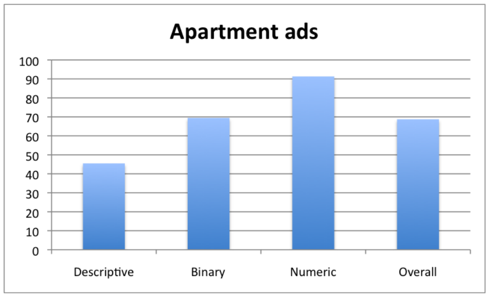

Figure 2: Total accuracy (F-measure) for apartment ads and under each category

Hence , a supervised method can be a better approach where the keys are assigned after being trained on an annotated training corpus. This will also eliminate the problem of *descriptive keys* not appearing in the text because such keys will appear in the training set as tags. In the next section, we describe a Conditional Random Field based supervised learning method to solve the same problem but with a new assumption that the set of keys for a given topic is known beforehand.

Supervised Learning

There are certain aspects of the data which were not properly considered in the unsupervised approach. Particularly, for the cases where the (descriptive) keys do not appear in the text (e.g. the key "color"). Applying a supervised method trained on a manually annotated training set solve this problem, as the unknown keys can be included as annotating labels.

The problem statement slightly changed in the supervised learning method but the end goal of this work essentially remain the same. The problem statement in the supervised approach is,

Given a sequence of words from the ads $\{x_1, x_2, ..., x_k\}$ what is the best hidden key sequence $\{y_1, y_2, ...y_k\}$ that describes the observed word sequence. In other words, what key sequence maximizes the conditional probability,

$$P(y_1, y_2, ...y_k|x_1, x_2, ..., x_k) \qquad (1)$$

Here, the set of keys $Y = y_1, y_2, ...y_n$ is known in prior. Once the sequence of words in an ad is automatically labeled by this model, the original ad can be converted into a tabular structure using the keys on one side and the corresponding words from the ads on the other, keeping it same with our original end goal.

The ads are usually written in an informal manner often using syntactically incorrect grammar, incomplete sentences and incorrect spellings. However, there is an inherent sequential aspect to these ads. If a word is assigned a label then the next word is more likely to have the same label. This property of the text was not taken into accaount in the unsupervised approach. There are other properties of the data, which the unsupervised approach did not consider, e.g. dealing with unknown words. If a particular word is not found in the training corpus the unsupervised method failed to classify it properly. All these properties make the problem similar to other NLP tasks such as POS-tagging or Named-entity recognition. Model like HMM, CRF have been quite popular in dealing with such NLP problems. Past works have shown that Conditional Random Fields (CRF) demonstrate better performance for these tasks compared to other models like HMM, Maximum Entropy (Sha and Pereira 2003)(Pinto et al. 2003). Considering this fact and the nature of the problem in hand, we decided to employ a CRF based approach for this problem.

Moreover, a variety of features can be included in a CRF based model. In our context, including a large number of features can accurately model the irregularities in our data. Particularly, linguistic features can help in dealing with ambiguity and unknown words. Also, it enables the use of previous words and keys as features, which can model the sequential aspect of the data. Implementing the model using CRF has other advantages, as well. CRF considers the entire key sequence while training. This results into optimizing the model parameters with respect to the entire key sequence. However, this makes the optimization slightly more costly but it increases the accuracy. Also, this approach introduces some options in building the model. Figure 3 shows how CRF can be used to annotate the ad from Table 1. From this figure we see that the keys are quite localized. They tend to occur next to each other. This property can be used as a feature to improve the accuracy. Across many ads, the sequence of keys also follow a pattern, i.e. users tend to talk about the model, color, mileage, features following a pattern. All these properties of the text make CRF a good choice for the task in hand.

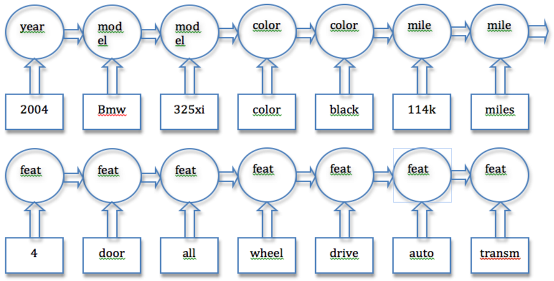

Figure 3: Linear chain CRF model to annotate ads (from Table 1) with keys

Conditional Random Fields

Conditional Random Fields (CRF) are discriminative probabilistic models (Lafferty et al., 2001) used for labeling sequential data. Given an input sequence of words from the ads $x = (x_1, x_2, ..., x_n)$ and an output sequence of keys $y = (y_1, y_2, ..., y_n)$, the model is defined as follows:

Table 4: Subset of labels used for car and apartment ads

Apartment Labels	Car Labels
location	model
address	year
size	price
contact	size
rent	feature
bedrooms	miles
baths	colour
kitchen	phone
floor	email
phone	engine
email	interior
	problems
	condition

$$P(Y|X) = \frac{1}{Z(x)} exp(\sum_{i}^{N} \sum_{k}^{K} \lambda_i f_i(y_{i-1}, y_i, x_i, i)) \tag{2}$$

where, $f_i(y_{i-1}, y_i, x_i, i)$ is the feature function comprising of current word, current key and the previous key. λ_i are the model parameters whose values are estimated from the learning process. $Z(x)$ is the normalizing factor and it is a function of x, as CRF computes the conditional probability instead of joint probability between x and y.

After the training process and parameter estimation, given an ad, the learned model tries to find a sequence of y_i which can best describe the observed sequence of words $(\mathbf{x})$ from the ad.

Training and Extraction of Label-Value Pairs

600 ads per topic were randomly selected from the corpus described in the Datasets section to be used as a training set. Each word in the training set was manually tagged with a label (some of the labels used is shown on Table 4). The ads contained some words which are mostly exclamatory and have limited relevance. These words are not required to be a part of the structured ad. Such irrelevant words in the ads were tagged using a special key "null". An additional 200 ads were selected and similarly annotated to be used as the test set.

Various features were used to build the model. Some of the features were textual, i.e. the words or some linguistic feature of the words and some were binary, expressing some properties of the words. whose value is either 0 or 1. A subset of the features used is shown in Table 5. These features were added in the model using the feature function of CRF $f_i(y_{i-1}, y_i, x_i, i)$ (Equation 2). The model was trained on the annotated set to learn the parameter λ_i. The learned value of this parameter was later used to get the key sequence for a new ad.

Two separate models were built for the two different topics, having the parameter sets Λ_{cars} and $\Lambda_{apartments}$. The learned models for a topic can be applied on a new unstructured ad on that topic to generate a labeled version of the ad. In the labeled ad, the keys can be extracted along with the corresponding words assigned to that key. The extracted pairs are presented as a table which is the structured form of the unstructured ad.

Experiments and Results

The model was trained on a training set of 600 ads. An additional 200 ads were annotated in the same way to evaluate the model. The size of the training and testing set was small due to the high cost of building such sets. The final results are reported based on the the accuracy computed on the test set of 200 ads [2]. In Figure 4 the X axis shows the performance for different experiments. We performed 10 different experiments where various combinations of features (including word window sizes) were used.

The baseline model is defined as where the key is solely dependent upon the current word $(f_i(y_i, x_i, i))$. This model gave an accuracy of 48.23% in the car ad test set. We increased the window size of words to include the previous word along with the current $(f_i(y_i, x_{i-1}, x_i, i))$. The performance increased to 56.75% and by making the window as (x_{-1}, x_0, x_{+1}) took the accuracy to 60.12%. Finally, experimenting with window sizes, we found the best performance (accuracy of 67.45%) with a window of $(x_{-2}, x_{-1}, x_0, x_{+1}, x_{+2})$.

We analyzed the error cases and added some more features based on some surface characteristics of the words. These features were mostly binary. Example of such features include, *is there a digit in the word*,*is there a symbol in the word* etc. Including the digit feature improved the performance by almost 0.6. However, the symbol feature did not add to the accuracy. Based on this observation, we added features involving common symbols, like, '\$', '-'. Inclusion of these features proved to be effective and the accuracy rose to 73.66%.

Finally, we included features which looked into deeper characteristics of the words. Using regular expressions, the features looked into whether the word looks like a phone no, email address, any unit (e.g. square feet/sq. ft. or cc/litre of car engines etc.) Experiments conducted with these features reported an accuracy of **74.07%**. Repeated experiments beyond this point did not improve the accuracy significantly. All the accuracies reported here based on car ad test set 4, which we used as the development set. The feature set which reported the highest accuracy in the development set, achieved an accuracy of **72.59%** on the apartment test set.The variation of accuracy with the performed experiments is summarized in Figure 4 and Table 6. A subset of the final feature set is shown in Table 5.

Table 5: A subset of the final feature set

Feature	Description
x_0	Current word
$x_{-2}, x_{-1}, x_{+1}, x_{+2}$	previous and following two words from the
	current word
digit in w_0	whether the current word has digits
'\$' in w_0	whether the current word has '\$' sign
'-' in w_0	whether the current word has '-' sign
phone pattern in w_0	whether the current word matches with a phone no. pattern
email pattern in w_0	whether the current word matches with a email id pattern

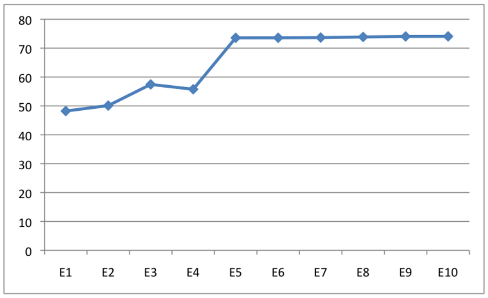

Figure 4: Accuracy for each experiment. There were 10 experiments performed on car ads with combinations of different word-window size and feature sets. Y-axis shows the accuracy in percentage.

Conclusion and Future Work

In this paper, we presented a solution to the problem of structuring unstructured online ads with a $< key, value >$ style representation. We proposed a graph-based unsupervised algorithm which gave a performance with an accuracy of 67.74% for cars and 68.74% for apartment ads downloaded from Craigslist. We also presented an alternative supervised learning algorithm where we used CRF to compute the most probable label sequence given an observed sequence of words in an ad. The supervised algorithm achieved an accuracy of 74.07% and 72.59% respectively for car and apartment ads. Lower accuracies in the unsupervised method can be attributed to the fact that there are some aspects to the problem which are very difficult to model in an unsupervised method. Implementation of the supervised algorithm actually shows that some of the shortcomings of the unsupervised method can be reduced by the supervised method.

We are currently exploring the possibilities to further enhance the accuracy of our algorithms. One possible ex-

Table 6: Accuracies for the supervised model for car and apartment ads

Cars	Apartment Rental
74.07%	72.59%

[2] All the accuracies presented in the section are for the car ads. Because, the car test set was used as the development set. A summary of the results is presented in Figure 4 and 6

tension to the unsupervised method can be to introduce some probabilistic learning and inference mechanism. Such a method can overcome the failure cases of the deterministic approach currently employed. For the supervised approach, the task described in this paper slightly differs with similar tasks such as, POS tagging and NER. In those tasks, every consecutive words are usually assigned different labels. In this problem, consecutive words are very likely to have the same label. Using some other model like Semi-markov CRFs can potentially increase the performance. Similarly, using a parser or a chunker can help to identify the consecutive but related words (e.g. “power windows”), which can avoid assigning different labels to each of them. Another approach to improve the accuracy for the supervised method can be to increase the annotated training set from 200 to a larger number. An alternative to this can be to explore the possibility of a semi-supervised (Sarawagi and Cohen 2004) approach: using a small amount of prior information on a given topic, can we further increase the accuracy? This method can reduce the cost of manually annotating huge number of ads, on the other hand perform better compared to a completely unsupervised approach. Finally, processing huge number of ads can be slow to process serially. Thus, adapting a distributed approach can improve the performance in terms of time. Storing the affinity graph using some distributed storage can help in parallelize the testing part and reduce time considerably.

Acknowledgement

This project was supported by the Center for Technology and Economic Development (CTED), New York University Abu Dhabi

References

Aone, C.; Okurowski, M. E.; Gorlinsky, J.; and Larsen, B. 1999. *A trainable summarizer with knowledge acquired from robust NLP techniques*. 71–80.

Blei, D. M., and Moreno, P. J. 2001. Topic segmentation with an aspect hidden markov model. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’01, 343–348. New York, NY, USA: ACM.

Conroy, J. M., and O’leary, D. P. 2001. Text summarization via hidden markov models. SIGIR ’01, 406–407.

Druck, G.; Mann, G.; and McCallum, A. 2008. Learning from labeled features using generalized expectation criteria. SIGIR ’08, 595–602.

Druck, G.; Settles, B.; and McCallum, A. 2009. Active learning by labeling features. EMNLP ’09, 81–90.

Evans, D. K.; Mckeown, K.; and Klavans, J. L. 2005. Similarity-based multilingual multi-document summarization. *IEEE Transactions on Information Theory* 49.

Grenager, T.; Klein, D.; and Manning, C. D. 2005. Unsupervised learning of field segmentation models for information extraction. ACL ’05, 371–378.

Haghighi, A., and Klein, D. 2006. Prototype-driven learning for sequence models. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL ’06, 320–327. Stroudsburg, PA, USA: Association for Computational Linguistics.

Kupiec, J.; Pedersen, J.; and Chen, F. 1995. A trainable document summarizer. SIGIR ’95, 68–73.

McCallum, A. 2005. Information extraction: Distilling structured data from unstructured text. *Queue* 3(9):48–57.

McKeown, K. R.; Klavans, J. L.; Hatzivassiloglou, V.; Barzilay, R.; and Eskin, E. 1999. Towards multidocument summarization by reformulation: Progress and prospects. AAAI ’99/IAAI ’99, 453–460.

Michelson, M., and Knoblock, C. A. 2008. Creating relational data from unstructured and ungrammatical data sources. *J. Artif. Int. Res.* 31(1):543–590.

Pinto, D.; McCallum, A.; Wei, X.; and Croft, W. B. 2003. Table extraction using conditional random fields. SIGIR ’03, 235–242.

Pradhan, S.; Hacioglu, K.; Ward, W.; Martin, J. H.; and Jurafsky, D. 2003. Semantic role parsing: Adding semantic structure to unstructured text. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM ’03, 629–. Washington, DC, USA: IEEE Computer Society.

Probst, K.; Ghani, R.; Krema, M.; Fano, A.; and Liu, Y. 2007. Semi-supervised learning of attribute-value pairs from product descriptions. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI’07, 2838–2843. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Radev, D. R.; Jing, H.; and Budzikowska, M. 2000. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic Summarization - Volume 4*, NAACL-ANLP-AutoSum ’00, 21–30. Stroudsburg, PA, USA: Association for Computational Linguistics.

Sailer, A.; Wei, X.; and Mahindru, R. 2008. Enhanced maintenance services with automatic structuring of it problem ticket data. *2013 IEEE International Conference on Services Computing* 2:621–624.

Sarawagi, S., and Cohen, W. W. 2004. Semi-markov conditional random fields for information extraction. In *In Advances in Neural Information Processing Systems 17*, 1185–1192.

Sha, F., and Pereira, F. 2003. Shallow parsing with conditional random fields. NAACL ’03, 134–141.

Svore, K. M. 2007. Enhancing single-document summarization by combining ranknet and third-party sources.

Wang, H.; Zhang, D.; and Zhai, C. 2011. Structural topic model for latent topical structure analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT ’11, 1526–1535. Stroudsburg, PA, USA: Association for Computational Linguistics.