Tracking the Occurrence and Academic Effects of Sexual Violence at a University

---

A Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment

of the requirements for the degree

Master of Science

by

Xiaoqian Liu

May

2016

APPROVAL SHEET

The thesis

is submitted in partial fulfillment of the requirements

for the degree of

Master of Science

_____

AUTHOR

The thesis has been read and approved by the examining committee:

Donald Brown

_____

Advisor

Matthew Gerber

_____

Kathryn Laughon

_____

_____

_____

_____

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, Dean, School of Engineering and Applied Science

May

2016

Abstract

Most of student sexual assault victims are unwilling to report to the police. This makes investigation and spatial temporal analysis less representative because of lack of documented incident addresses. Due to this under-reporting issue, we are motivated to discover additional sexual assault incidents outside of the local crime report, for instance emergency records and news reports. These addresses can be quite useful to discover some undiscovered patterns in the crime analysis. In this research, I present an approach to automatically extract street addresses from news reports by applying a sequential labeling technique and semi-supervised learning. The previous work on address extraction only focuses on web pages where addresses are separated from other texts; however our problem needs to retrieve addresses embedded in texts. We built the Gradient Boosting and Conditional Random Field (CRF) models to solve this problem. In addition, we utilized a semi-supervised learning algorithm to use additional unlabeled data to further improve the predictive performance. In the end, we compared the patterns of extracted addresses from documented addresses in the crime report.

Acknowledgement

I would like to thank my advisor Prof. Brown for his guidance during the last two years. His data mining class made me become curious about Machine Learning. I have learned a lot from him. I would thank Prof. Gerber and Prof. Laughon on the thesis committee for providing insightful suggestions and comments. I would like to thank Phill Trella and the Office of Graduate and Postdoctoral Affairs for accepting me to the Presidential Fellowship in Data Science, which provides funding and assistance for this research. I would like to thank Colleen Sanders for collaborating with me on this project.

Also I want to give thanks to Prof. Beling, Prof. Gu and every other professor teaching in this department. I am grateful for achieving both undergraduate degree and master's degree in this department. Thanks for teaching me knowledge and broadening my vision over the last six years.

Lastly, I want to thank my parents, Mingye Wang and Wangting Liu for their unconditional love and support. Thank you everyone who helped me. Without your help I could not make this far.

Table of Contents

List of Figures

List of Tables

Section 1: Introduction

College aged women are at highest risk for sexual violence and Intimate Partner Violence (IPV). Most female victims of rape (78.7%) experience their first victimization before the age of 25 with 38.3% experiencing their first rape between ages 18-24. IPV has similar patterns with 71.1% of women experiencing their first form of IPV before age 25 and 47.9% between ages 18-24 (Breiding, et al., 2014). Women who experience physical violence are at risk for revictimization whether the event occurred before or during college (Smith, White & Holland, 2003).

Student victims are even less likely to report the incidents to police (20%) compared to non-students (32%) (Sinozich & Langton, 2014). In at least one study, 51% of women discussed abuse with their health care provider (Morse, Lafleur, Fogarty, Mittal & Cerulli, 2012). The Centers for Disease Control and Prevention (CDC) identifies sexual violence and intimate partner violence (IPV) as public health problems with significant negative impacts on the physiological, emotional and social well-being of victims both in the acute phase following the violent event and chronically afterwards (Breiding, et al., 2014).

In our previous research, we analyzed sexual assault incidents in Charlottesville between 1990 and 2015 recorded in a crime report provided by local police (Clougherty et al, 2015). According to the kernel density plots (Figure 1), sexual assault incidents are more likely to happen during the midnight compared with other times of the day. The hotspot center is relatively fixed in the Main Street and Corner area. There more incidents in the area during the week and summer time.

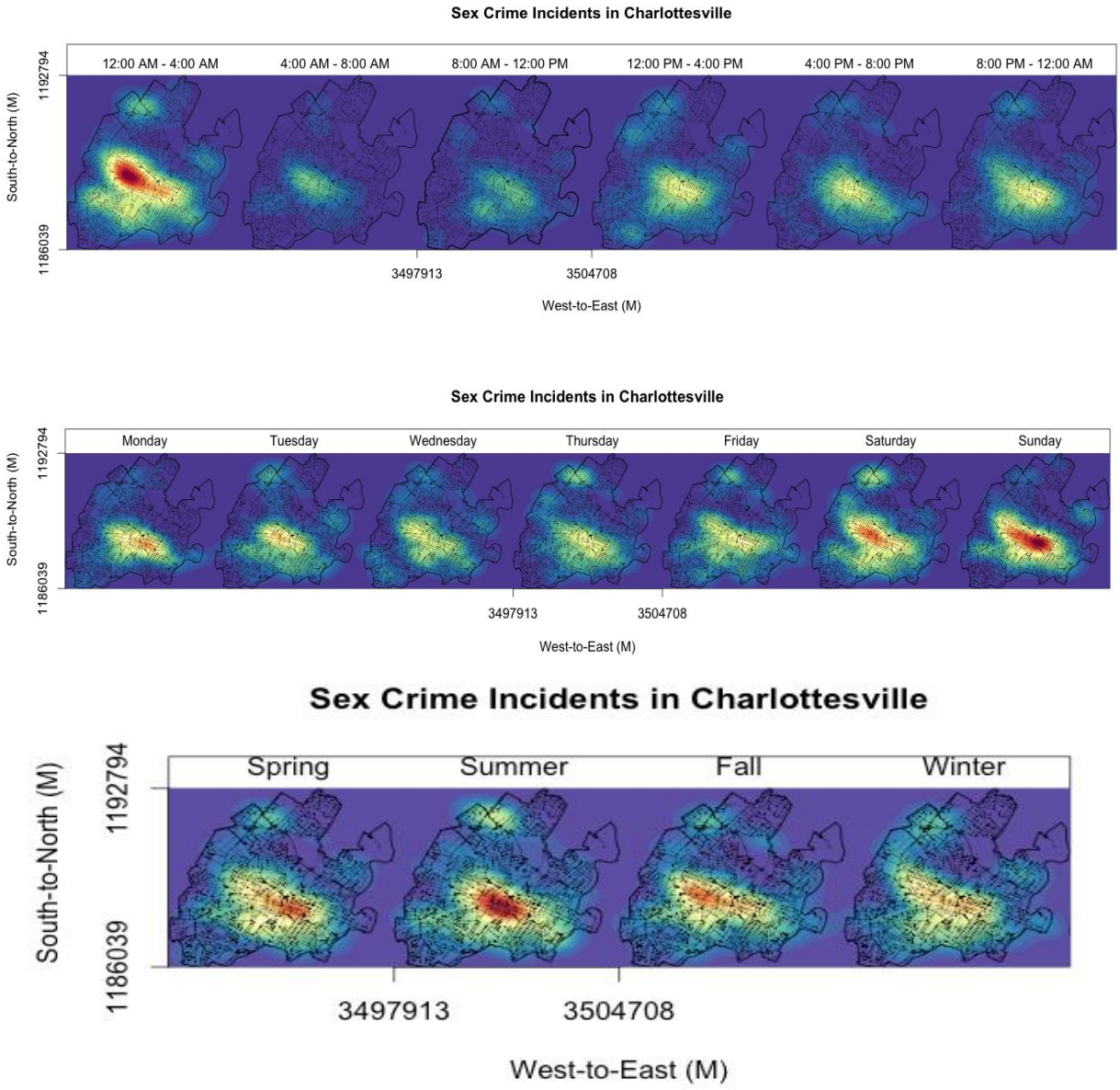

Figure 1. Kernel Density Plots of sex crime incidents in Charlottesville

We found that the proximity to Greek houses, downtown entertainment area, hotels and retails are strongly correlated to the occurrence of incidents. These findings were consistent with previous work on exploring the correlation between sexual assault and Greek culture and life.

Moreover, temperature is the most statistically significant weather factor in the Random Forest model as well as the logistic regression model. This is also indicated by the kernel density plot in Figure 1.

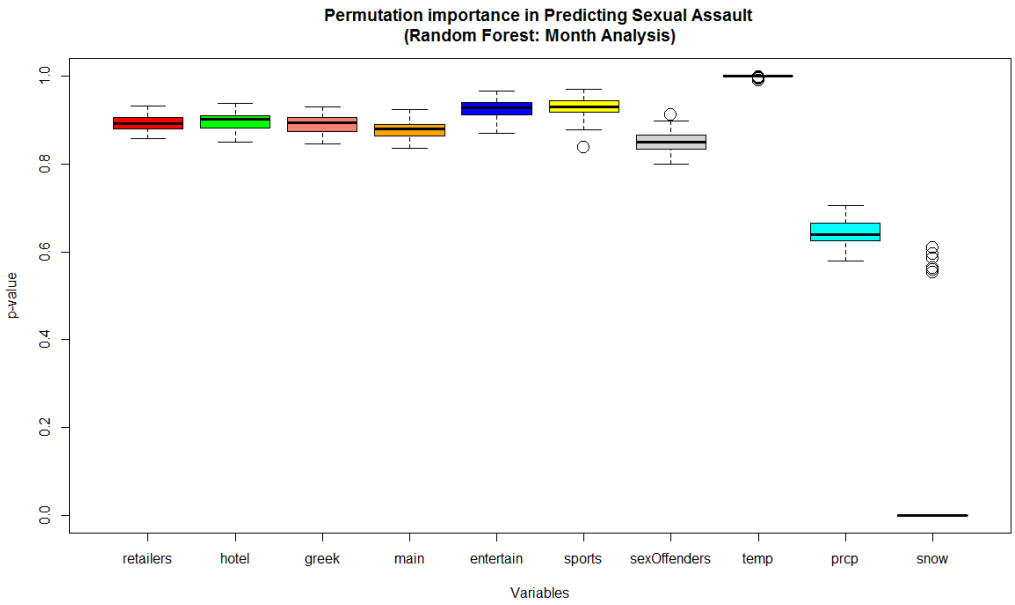

Figure 2. Permutation Importance of Random Forest Model for Monthly Analysis

In order to find addresses not covered in the police report, this research aims to extract addresses from news reports containing keywords related to sexual assaults. The following major challenges make the problem difficult. First of all, there does not exist any existing manually labeled news reports data sets. Also, news reports cannot be automatically labeled without human supervision. Even though there are some previous work on address extraction from web pages, those addresses are in specific address blocks which are separated from other texts. These data can be automatically labeled based on some particular HTML structure information. Secondly, similar to the problem in address extraction from web pages (Yu, 2007), the same address may have various forms in the news reports. For example, the author may neglect the

street number, use street abbreviations instead of full names and misspell the names. Therefore, it is difficult to generalize templates for extracting addresses.

The thesis discusses about the details of techniques and models in related work in Section 2. We examined the performance and characteristics of existing solutions for solving address extraction problem and Name Entity Recognition. In Section 3, we demonstrate details and sources of data in this project. Section 4 includes explanation of data preprocessing, feature extraction and multiple model candidates for classifying address entities, followed by experiment and matching results in Section 5. Lastly, we compare performance of different models with and without semi-supervised learning and make conclusion in Section 6.

Section 2: Literature Review

This section provides the background context and contribution of this work. Section 2.1 summarizes main techniques used in Name Entity Recognition. Section 2.2 discusses about address extraction techniques in related work. Section 2.3 includes the state-of-art keyword extraction applications.

2.1 Name Entity Recognition

Name Entity Recognition (NER), coined in the Six Message Understanding Conference (MUC-6) is an information retrieval task which aims to identify information units such as names of people, organizations, locations, time, date and many other entities (Ratinov, Roth, 2009). In the early work, NER task was considered as a 'proper names' with three major specializations, namely 'Persons', 'Locations' and 'Organizations' (C. Thielen 1995). They are also known as 'Enamex'. For 'Location' entities, they can be further categorized into fine-grained addresses such as city, state and others (Nadeau, 2007). Street address extraction in our research belongs to 'Enamex' class of problems. Since this task contains much more complicated and varied patterns, direct application of existing NER systems can hardly solve our problem.

2.2.1 Supervised Learning

NER is mainly solved as a sequential labeling problem by the following supervised learning approaches: Hidden Markov Model (HMM) (Bikel, 1997), Decision Trees (Sekine, 1998), Maximum Entropy Models (MEMM) (A. Borthwick, 1998), Support Vector Machines, Conditional Random Field (CRF) (McCallum and Li, 2003). These methods require a large

number manually labeled data set for training and define disambiguation rules from selected features (Sekine, 2007).

HMM is a generative model which classifies input sequence. For the NER problem, the name entity types are considered as hidden states and input sequence represents observed states. The objective of HMM is to discover the most probable hidden state sequence via Viterbi Algorithm. The earliest HMM was developed by Bikel, which achieved 93% F-score in a news data set (Bikel 1999). Based on his model, Zhou and Su improved the HMM NER tagger by introducing external features which achieved 96.9% and 94.3% F-Score in MUC-6 and MUC-7 English Name Entity tasks (Zhou, Su, 2002).

In 2001, CRF was introduced as a sequence modeling framework that possess the advantages of MEMMs (Lafferty, et al, 2001). It has an exponential model for the joint probability with respect to the entire sequence of labels given with the input sequence. This allows to trade off weights of different features at different states. More details are documented in Section 4.3.3.

2.2.2 Semi-supervised Learning

Yarowsky stated a powerful property of human language called 'One Sense per Collocation': a target word and its neighbors in the in the same discourse are very likely to share a common meaning (Yarowsky, 1993). Under this assumption, Yarowsky algorithm can learn

The Co-training algorithm assumed that features in the same classifier can be divided into two independent sets (Blum and Mitchell, 1998). Each of them is capable for classification. In every iteration of the semi-supervised learning process, each set of features is exploited

alternatively to classify unlabeled corpus and data assigned with high confidence is inserted into the training set.

More recent work on semi-supervised learning applied the structural learning on finding a shared predictive structure from a large amount of automatically generated auxiliary classification problems (Ando, Zhang, 2005). They proposed an algorithm called ASO-semi which achieved F1-score 89.31 and 75.2 in CoNLL'03 English and Germany data sets which outperformed previous semi-supervised learning systems.

2.2.3 Unsupervised Learning

Since the annotated data sets might not be available in every discipline and language, unsupervised learning approach becomes a great candidate. Etzioni, et al proposed KnowItAll system without domain-specific knowledge in an unsupervised and scalable manner (Etzioni, et al, 2005). The system consists of three major components bootstrapping, extractor and accessor. It starts with bootstrapping a small set of rules and discriminators of each predicate and then iteratively uses extractor to extract rules and appends extracted lists (e.g. constraints and keywords) and discriminators by accessors. It utilized pattern learning, subclass extraction and list extraction methods to improve the recall.

2.2 Address Extraction from Web Pages

Addresses extraction from documents is one of information retrieval tasks which can bring us a lot convenience. This section covers major address extraction methods, namely pattern-based and machine learning methods.

Pattern-based method mainly utilizes Gazetteer and Regular Expressions with some heuristic rules. Gazetteer is a geographical dictionary which can provide a lot of important information such states, cities and streets. By using these geographic specific indicators and Gazetteer lookup, this approach achieved F1 score of 75.3% which outperforms than the regular expression method (Yu, 2007).

Another popular approach is using machine learning models. In Yu's research, he applied decision tree models as well as the pattern based methods mentioned in the previous section. This method first tokenizes a web page document and categorizes them into one of four classes, namely Start, Middle, End and Others. After the classification step, he retrieved addresses based on the label sequence. According to his evaluation results, the decision tree + regular expression method has the best performance. Its precision value is 95.2% and recall value is 81.1%. Built on his work, Chang and Li created MapMarker, a new system for extracting postal addresses and associated information (Chang, Li, 2010). They applied ANNIE annotation and word segmentation based on address keywords (such as street suffix) in the preprocessing stage. They exploited the CRF model for classification and improved the prediction accuracy to 91% F1-Score in comparison with Yu's model.

2.3 Contribution

In addition to the n-gram features and machine learning models mentioned from previous work in address extraction, this research also explores some additional features, such as word distance to keywords and use NER entity type of each n-gram and apply semi-supervised learning to iteratively self-train the model. By using 1000 additional unlabeled data to iteratively improve

the baseline model in this approach, we improved the average weighted F1-Score from 72% to 79%.

Section 3: Data

In order to prepare training data for this research, we extracted news articles published on Washington Post and Cavalier Daily from LexisNexis, by using keywords including ‘sexual assaults’ and ‘Street’ and other similar terms. In total, there are 374 Washington Post reports and 731 Cavalier Daily articles which were manually labeled. Only 237 addresses exist in the Washington Post data set and 132 addresses exist in the Cavalier Daily data set. Cavalier Daily is a student-operated news media and it is the oldest daily newspaper in Charlottesville area. We assume that their reports in combination with Washington Post may have a good coverage of news articles in Charlottesville and nearby Virginia area. We also collected over 500 reports for both data sets as unlabeled data used for augmenting our training classifier under the semi-supervised learning setting.

Table 1. News Reports Data Overview

Source	Count
Washington Post	374 Documents, 237 Addresses
Cavalier Daily (University Wire)	731 Documents, 132 Addresses
Unlabeled	998 Documents
All news reports from 1990-2015 containing keywords (e.g. Charlottesville + sexual assault related words)	213 Documents

From our previous research, we received a list of 852 sexual assault incidents with location, time and incident type (Table 2) from January 1990 to January 2015, in the courtesy of Mr. Cody Bowman from the Charlottesville Police Department.

Table 2 Crime Report Data Overview

Incident Type	Count
Forcible Fondling	345
Forcible Sodomy	107
Rape	389
Sexual Assault with an Object	11
Total	852

By plotting number of incidents in each year (Figure 3), we found that only 9 incidents happened during 1990 - 1996. From 1991 to 1994, there is even no documented sexual assault. This makes us become interested in finding incidents from news reports in this time window

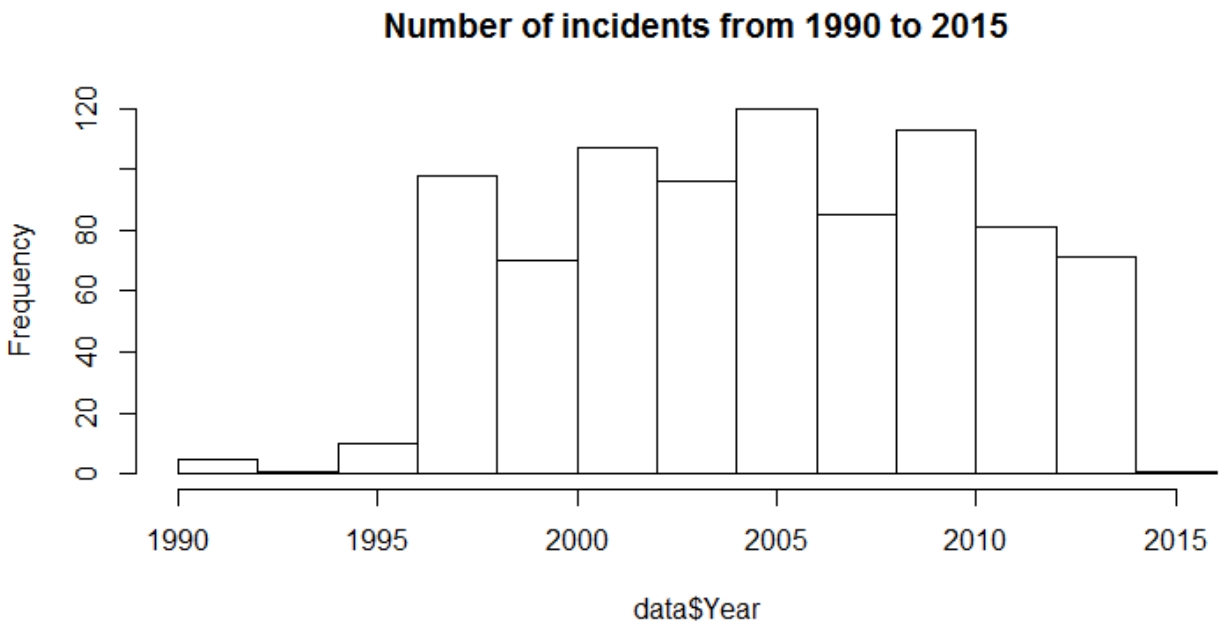

Figure 3. Histogram of number of incidents from 1990 to 2015

In addition, we will use these addresses to match with extracted addresses related to sexual assaults from news reports from 1990 to 2015. Figure 4 illustrates a visualization of addresses on

the top of Charlottesville map. We will generate visualizations for extracted addresses and identify discrepancies in the addresses patterns.

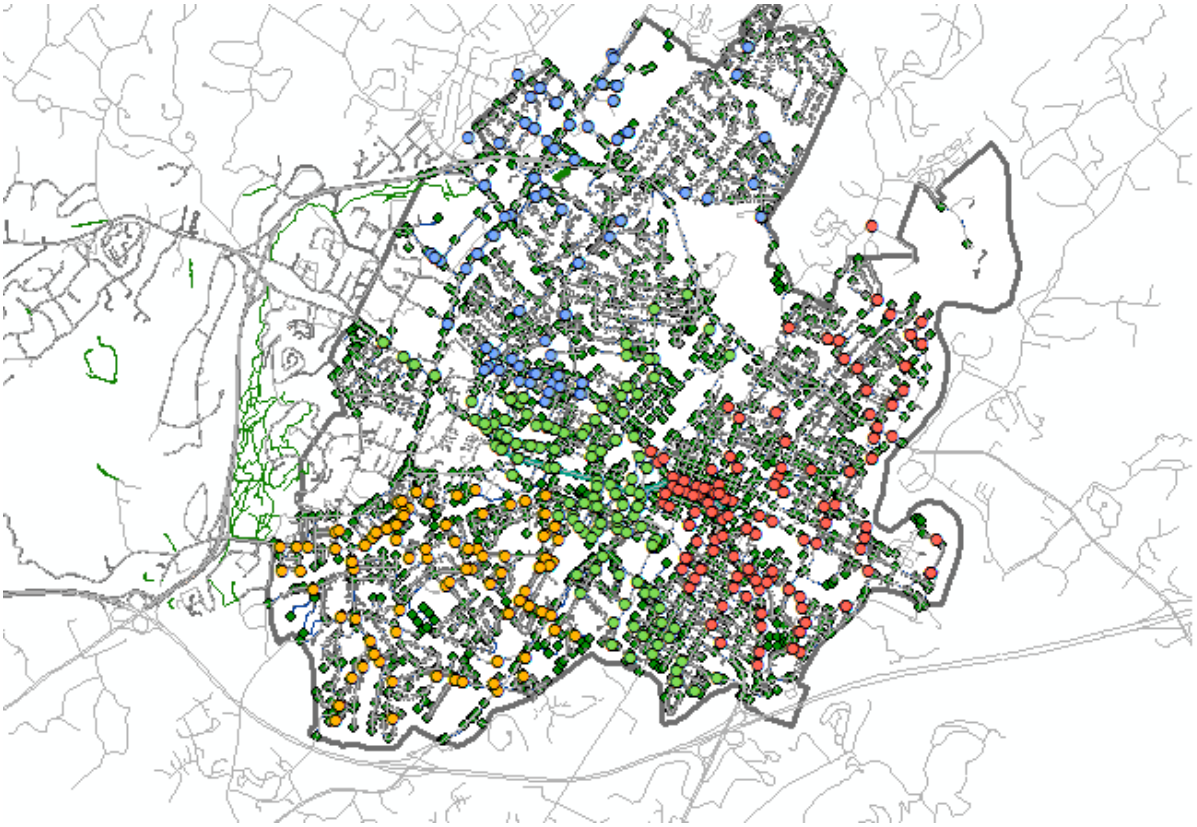

Figure 4. Cluster plots of sexual assault incidents from 1990 to 2015.

Section 4: Methodology

This section explains the methods for address extraction in news reports. Figure 5 demonstrates a high-level summary of address extraction method. Section 4.1 describes how to preprocess the news documents. Section 4.2 lists the features used in this model and discusses about the details of feature extraction. Section 4.3 describes the model candidates and Section 5 shows an algorithm to retrieve addresses based on the class labels.

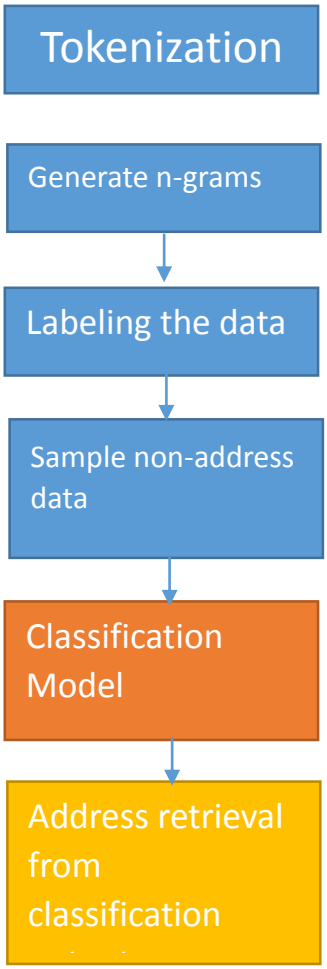

Figure 5. Summary of address extraction from news reports

4.1 Preprocessing

4.1.1 Class Labels

First of all, each news report document needs to be tokenized. We first split the documents into sentences and utilize regular expression tokenizer to tokenize each sentence. We kept punctuation since we want to use it as a feature later. Then we created n-grams by concatenating tokens in a context window size of (n-1)/2. For instance, given with a sentence "On 830 Mckeet street.". When n=1, it can be transformed into three different sequences, "on-830-mckee", "830-Mckee-Street" and "Mckeet-Street-.". For this project, we set the context window size as 1.

By adopting the BIEO tagging, there are four different classes in our model: beginning of the address (B), inside of the address (I), end of the address (E) and outside of the address (O) (Yu, Chang). The class label for each n-gram is determined by its central word. For instance, "830-Mckee-street" is considered as class I, since Mckee is the second word of the address.

Table 3. Class Labels

Class Label	Description	Example
B	Beginning of the address	**On 830** Mckee
I	Inside of the address	830 **Mckee** Street .
E	End of the address	Street **.** One
O	Outside of the address/Others	**. One** incident

4.1.2 Data Sampling

After tokenizing the documents and labeling them based on BIEO tagging, we calculated total number of n-grams for each class in order to have an overview of data distribution. From Table 4, we can observe that around 100 out of 140,629 tokens belong to addresses, which shows an

extremely disproportionate class distribution. The excessive amount of ‘O’ classes not only waste computation power but also barely improve the predictive performance. Therefore, we decided to randomly sample sentences which are only consist of “O” classes. By experimenting different sample rate and comparing their predictive performance on the test set, we found that 1/3 of number of address classes is the best sample rate which causes the highest F1-score.

Table 4. Number of examples in each class in training set

	B	I	E	O
Washington Post	59	30	53	140,629
Cavalier Daily (University Wire)	109	47	86	224,991

4.2 Feature Selection

4.2.1 Word-Level Features

For each word in a bigram, we collected the following 15 features in Table 2. We extracted street suffixes and their abbreviations from United States Postal Services website and saved them to a dictionary. We also prepared a set of directions and ordinal indicators for feature lookup. For each word, we utilized dictionary lookup and regular expression method to collect feature values. Also, we applied 7-class NER in the preprocessing stage by using Stanford NER tagger. The seven classes are location, person, organization, money, percent, date and time. Compared with the previous work, we added some additional features such as the minimum word distance to name entities and minimum word distance to the keywords (i.e. Sexual Assaults and its synonyms).

Table 5. Feature Description and Examples

Feature Description	Example
First letter is capitalized	Mckee Street
The word is a street abbreviation	St, Ave, Street, Avenue …
All characters are in uppercase	DC
All characters are in lowercase	we
The word is an ordinal indicator	st, rd, th
The word contains AM/PM	10am
The word contains time	10:00
The word contains time zone	PT, CST, PST, ET
The word is a direction	NW, N, W
The word contains a punctuation	R&D
The word is a digit	1
The word is a combination of digits and letters	12th
The word is an email address	XX@XX
Word length	5
First letter is a letter	A
Name Entity Recognition	'ORGANIZATION', 'LOCATION'
Word Distance to keywords (i.e. 'sexual assault', 'rape')	6
Word Distance to extracted Name Entities	6

4.3 Modeling

4.3.1 Gradient Boosting

In this research, we utilized XGBoost package in Python for generating models. XGBoost stands for "Extreme Gradeient Boosting" and is a scalable and computationally-efficient implementation of Gradient Boosting model (Friedman, 2001). For tree boosting in general, it has the following regularized learning objective.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$$

Given with a data set which contains n exmaples and m features, the boosting model uses the summation of k feature functions, f_k to predict the output. In this formula, $\mathcal{F} = \{f(x) = w_{q(x)}\}(q: \mathbb{R}^m \longrightarrow \mathrm{T}, w \in \mathbb{R}^T)$ represents the space of regression trees. q is a tree structure composed of a set of decision rules. It maps an example to its corresponding leaf index. Each f_k has its associated leaf weights w and a tree structure q. The following equation indicates the regularized learning objective of the collection of feature fucntions.

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{t=1}^{t} \gamma T + \frac{1}{2}\lambda\|w\|^2$$

The first term, $l\left(y_i, \hat{y}_i^{(t)}\right)$ is a differentiable convex loss function measuring the prediction loss of y. The second term is an L2-regularization term to avoid overfitting. In order to effectively optimize the formula, the previous formula can be turned into the following equation using additive training, namely boosting:

$$\mathcal{L}^t = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + (\gamma T + \frac{1}{2}\lambda\|w\|^2)$$

$\hat{y}_i^{(t)}$ is the prediction of the ith example at tth iteration. The model is iterativley improved by greedily adding $f_t(x_i)$. The optimization can be speeded up by applying the second-order Taylor Expansion. The individual tree structure q is created in a greedy approach which iteratively adds branches based on loss reduction values.

4.3.2 Principal Component Analysis (PCA)

PCA is a dimension reduction algorithm which maintains the most variance of the original data. The data is transformed into a set of orthogonally independent variables as known as principal components, which are linear combinations of variables. The first principal component represents the maximum magnitude of variance. Selecting a number of principal components less than total number of variables in the data set can achieve dimension reduction. It is usually solved by the Singular Value Decomposition (SVD).

4.3.3 Conditional Random Fields

Conditional Random Fields (CRF), an undirected graphical model is widely used in Part-of-Speech Tagging, Name Entity Recognition and many other Natural Language Processing tasks. Based on its definition, X, a random variable is a collections of input sequences, and Y is a random variable over label sequences. CRF is a random field globally conditioned on X. Its formula is as following:

$$p(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)\right)$$

$$Z(x) = \sum_{y} \exp(\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t))$$

In the equation, Z is the normalization factor, λ_k is an array of weight parameters and f_k represents real-valued feature function which can be in various forms (e.g. prefix of x_t and features of surrounding words). The structure of linear chain CRF is very similar to the Hidden Markov Model (HMM), shown in Figure 6. The major difference is that CRF is an undirected

model and it models on conditional probability distribution p(y|x) instead of joint distribution p(y,x). Compared with HMM which assigns the same score for the transition between y_t and y_{t-1} regardless the input xt, CRF takes into consideration of features of input sequence X as a discriminative model.

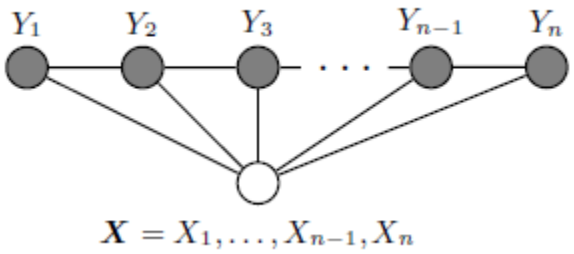

Figure 6. Linear Chain CRF over input sequence X.

Parameter estimation in CRF is solved by penalized maximum likelihood (McCallum, Sutton). The conditional distribution is modeled by the conditional log likelihood:

$$l(\theta) = \sum_{i=1}^{N} \log \mathrm{p}(y^i|x^i)$$

$$l(\theta) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k\left(y_k^i, y_{t-1}^i, x_t^i\right) - \sum_{i=1}^{N} \log Z\left(x^i\right) - \sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}$$

$l(\theta)$ is the result of substituting p(y|x) shown in the above and adding a regularization term $\sum_{k=1}^{K} \frac{\lambda_k^2}{2\sigma^2}$. Some optimization techniques for solving $l(\theta)$ are BFGS, a quasi-Newtown method (Bertsekas, 1999) and conjugate gradient on the approximation of the second-order information (Lafferty et al, 2001).

In this research, we applied the linear-chain CRF for prediction by using sklearn-crfsuite, a Python implementation CRFSuite (Okazaki, 2007). The reason of using this package is that it is convenient to customize feature functions and fast implementation. We also considered the PyStruct package, but we did not use it since it cannot generate class probability which prevents us from using this model in the semi-supervised learning setting and combining it with other models.

4.3.4 Semi-supervised Learning

Semi-supervised learning is quite useful when generating labels for the test set requires a lot of human efforts. For our project, we have to manually label addresses from news reports which limit the size of our training and testing data. Additionally, most of the news reports with keywords ‘sexual assault’ and ‘Charlottesville’ does not necessarily contain addresses. This makes the number of positive examples (i.e. addresses) even smaller. By taking the semi-supervised learning approach, we can use a large amount of unlabeled reports to help improve prediction accuracy of our model. In this section, we introduce a semi-supervised learning approach applied in this research: a simple semi-supervised learning approach applied in this research, proposed by Liao and Veeramachaneni (2009). In essence, the algorithm (Table 6) suggests to iteratively use a classifier trained by a training set to label each partition of unlabeled data, and fit the classifier again on the extracted new labels shown in Table 4.

Table 6. Semi-supervised NER algorithm by Liao and Veeramachanei

Given:
L - a small set of labeled training data
U - unlabeled data
Loop for k iterations:
Step 1: Train a classifier C_k based on L;
Step 2: Extract new data D based on C_k;
Step 3: Add D to L;

4.4 Address Extraction

Once class label prediction of each bigram is finished, we will precede to retrieve addresses which has the following possible sequences of labels: '0-1-2', '0-1-1…-1-2', '0','1', '2', and '0-1'. These sequences will be embedded in all the other irrelevant words marked as 'Others'. When an address is found, we will examine if it is in a valid format. Details of algorithm are outlined as below (Table 7).

Table 7. Algorithm Retrieval

Algorithm 1 Address Retrieval

for each predicted output **do**

if predicted class = 3 and predicted class of previous n-gram = 3

if an address candidate is found:

if the last word in the address is valid (direction/street abbreviation)
Output the address

Else if the second last word is valid
Output the address excluding the last word

```
else
        if  a word is not a punctuation and different from the last word of the address
                append this word to the address candidate
        else
                create a new address candidate
```

Section 5: Experiment Results

We measure the performance of models based on the precision, recall and F1-score of classification. We define precision as a ratio between the number of correctly identified class I and identified class I. Recall is a ratio between number of correctly classified class I and number of correct class I. F1-score is the major metric for evaluation, which is a combination measure of precision and recall. Consistent with the evaluation in CoNLL, incomplete identifications of spans (equivalent as the class in this project) are neglected and false positive errors are not penalized.

$$Precision(class\ I) = \frac{number\ of\ correctly\ identified\ class\ I}{number\ of\ identified\ class\ I}$$

$$Recall(class\ I) = \frac{number\ of\ correctly\ identified\ class\ I}{number\ of\ correct\ class\ I}$$

$$F1 - score(class\ I) = \frac{2 * Precision(class\ I) * Recall(class\ I)}{Precision(class\ I) + Recall(class\ I)}$$

5.1 Evaluation Results

We built the CRF model, XGBoost model with and without PCA on Washington Post and University Wire data sets separately. Since CRF model requires the features of n-grams grouped by sentences, we first splits the entire data set into sentences and then append features of each n-gram of each sentence into the same array. In order to compare the performance of all of the models on the same training and test data, we randomly sampled 70% sentences for a training set and 30% sentences for a testing set. For the XGBoost models, we concatenate features of each sentence in order to transform it into a $52 \times m$ feature matrix (m indicates number of n-grams).

The parameters for CRF model, L1 and L2 regularization variables are tuned in 10-fold cross-validation of the 70% training set. The model is optimized by LBFGS with L1 and L2 as 0.1 and set all possible transitions as true. Table 8 shows the precision, recall and F1-score values of the CRF model on the test set. The support value is equivalent to the class size. Even though we sampled a small amount of class 'O', the class size is still much greater than other classes. From the results, we can see that the CRF model achieved high precision and recall for almost each class, except the class E.

Table 8. CRF on training set of Washington Post

	Precision	Recall	F1-score
B	0.95	0.87	0.91
I	0.98	0.88	0.93
E	0.53	0.50	0.52
O	0.99	0.99	0.99

For the XGBoost, we defined the max depth of trees as 6, eta value as 0.1 and number of iterations as 250. The parameter values were chosen by grid search in 10-fold cross-validation of the training set. The results of XGBoost model are documented in Table 9. The average F1-score of XGBoost model across all the classes are quite similar to the CRF model. XGBoost model did a better job at identifying class E than the CRF model.

Table 9. XGBoost on training set of Washington Post

	Precision	Recall	F1-score

B	0.91	0.90	0.91
I	0.95	0.87	0.91
E	0.71	0.62	0.67
O	0.99	0.99	0.99

After applying PCA with 35 principal components, we run the XGBoost model again and generated the following results in Table 10. The number of principal components was selected from a grid search in the range from 25 to 50. The F1-score for class I is improved while the score for class E drops. In general PCA does not significantly improve the prediction.

Table 10. XGBoost + PCA on training set of Washington Post

	Precision	Recall	F1-score
B	0.91	0.91	0.91
I	0.95	0.85	0.90
E	0.64	0.44	0.52
O	0.99	0.99	0.99

We repeated the previous process on the University Wire data set. We tuned the CRF model parameters again in 10-fold cross-validation. Now L1 value become 0.5 and L2 value becomes 0.1.The model performance is shown in Table 11. From the results, we can see that the overall F1-score is worse than that of any model on another data set. Table 12 and table 13 show the results of XGBoost model and XGBoost model with PCA transformation respectively. From the results of the three models, we can conclude that CRF model had the best average F1-score

across three different classes and it has stable performance in both data sets. XGBoost model achieved the second best performance. Its precision value for class B and I are better than in CRF model. Therefore, an ensemble of these two models may provide a better performance. However, XGBoost model + PCA (20 principal components) did not work well on this data set. It generated the worst predictions compared with other models.

Table 11. CRF on the training set of University Wire

	Precision	Recall	F1-score
B	0.81	0.92	0.86
I	0.80	0.73	0.76
E	0.85	0.91	0.88
O	0.99	0.98	0.99

Table 12. XGBoost on the training set of University Wire

	Precision	Recall	F1-score
B	0.86	0.86	0.86
I	0.50	0.09	0.15
E	0.84	0.80	0.82
O	0.98	0.99	0.98

Table 13. XGBoost + PCA on the training set of University Wire

	Precision	Recall	F1-score

B	0.87	0.92	0.89
I	0.80	0.36	0.50
E	0.85	0.73	0.79
O	0.99	0.98	0.98

Since the performance of each model on the University Wire data set is worse than another data set, we experimented the semi-supervised algorithm mentioned in Section 4.3.4 by using 1000 unlabeled news reports. We start with the CRF model trained on the Washington Post data set which gives the following predictive results on the University Wire data set in Table 14.

Table 14. CRF model with labeled data

	Precision	Recall	F1-Score
B	0.85	0.54	0.66
I	0.80	0.78	0.79
E	0.82	0.68	0.74

We first transformed 1000 unlabeled reports into a collections of features of sentences. Then we partitioned them into 40 partitions. We followed the algorithm in Section 4.3.4 by iteratively applying the model on the unlabeled data and calculating the class probability on each partition. In each iteration, we retrieved the classes with probabilities higher than 0.6 and sampled sentences of class O since we also need negative examples. The results are shown in Table 15.

Table 15. CRF model with unlabeled data after applying semi-supervised learning

	Precision	Recall	F1-Score
B	0.82	0.89	0.85
I	0.76	0.83	0.79
E	0.77	0.64	0.70

The predictive accuracy of each address class, B, I, E is related to the percentage of correctly extracted addresses. According to the address extraction algorithm, it starts to collect a temporary address when a component of the address is detected until a non-address element is found. Intuitively, the beginning of the address (class B) is the most significant span since it is the initial condition of the algorithm. When the F1-score of class B is high, the algorithm can more easily identify the existence of an actual address. Otherwise, the algorithm is more likely to neglect the existence of an address and may reduce the accuracy of address extraction. When F1-score of class E is high, then the algorithm can correctly captures the ending component. Nevertheless, when class E and class I labels are mixed up, the algorithm still works since it terminates when a non-address element is visited.

At the beginning of the project, we also utilized several Regular Expressions patterns for address extraction. Nevertheless, its accuracy is much lower than using the machine learning approach. The result is consistent with the observations of using regular expressions in Yu's research (2007).

5.2 Matching

In the Data section, we briefly introduced a sexual assault incident report provided by the Charlottesville Police department. The report contains 852 addresses from 1990 to 2015. In order to address the potential under-reporting issue, we applied the CRF model with the semi-supervised learning algorithm mentioned in the last section and address extraction algorithm on the news data set (213 documents). This data set covers contents relevant with sexual assaults happened in the same time duration in Charlottesville. Table 16 lists out the manually labeled addresses in the news data set which roughly match with addresses in the police report. In total, there are 325 addresses which contain partial patterns with addresses in the police reports. For example, 'Swanson Drive' and '100 Swanson Drive' is a matching pair since the former one is included in the second one. The rough match rate is 38%.

Table 16. Addresses relevant with sexual assaults in Charlottesville

Address	Count	Address	Count
Swanson Drive	3	Preston Avenue	17
Fourth Street NW	9	Emmet Street	21
Second Street NW	3	Water Street	14
Monticello Road	13	Rugby Road	18
University Ave	6	Jefferson Park Avenue	20
Chesapeake Street	1	15th Street	9
Madison Avenue	5	Grady Avenue	5
Ivy Road	6	Gordon Avenue	1
High Street	12	Wertland Street	15
Elliewood Avenue	3	14th Street	10
Main Street	36	10th Street NW	3
5th Street	27	Sunset Avenue	1
Old Lynchburg Road	4	E Main St	19
4th Street	9		

Apart from the rough matches, we found 21 exact address matches shown in Table 17. Compared with the original block addresses in the police report, the extracted addresses have street numbers which provide even more specific and accurate geographic locations. In the news data, we also discovered some addresses are actually belong to the block address, however their patterns are slightly different than the general street address pattern, i.e. '500 block of Madison Avenue'. Therefore they are not included in the extracted address list.

Table 17. Exact matches

Address	Block	Count
Elliewood Avenue		3
105 Lankford Ave	100	1
Madison Avenue		1
221 E Main Street	200	7
601 Preston Avenue	600	2
117 5th Street SE	100	2
319 E Main Street	300	3
209 Monticello Road	300	1

In addition, there are 6 addresses not included in the police report. In order to validate whether these addresses are related to a sex crime incident, we went through all the documents which contain the following addresses and browsed their topics. Table 18 shows the addresses and their corresponding topics. Only 2 addresses are relevant with sex crimes: Stone Creek Ln is an address of a rape incident in Charlottesville; 16th Street is a location where a women got attacked before sent to the hospital. Other addresses are found in a court document with a sexual assault related keyword, a report of wanted on sex offenders and a report of a burglary incident.

Table 18. Addresses excluded by the police report

Address	Topic
900 South Street	Residence information contained in a court document
Stone Creek Ln	Rape incident in an apartment in Stone Creek Ln
16th Street	a woman was attacked in the 8500 block of 16th Street
West Market Street	Wanted on sex offender's last known address
Solomon Rd	Wanted on sex offender's last known address
Allied street	a burglary at Rocky Top Gym on Allied Street

We also investigated the performance of applying the CRF model trained on all the labeled data sets (Washington Post + University Wire) and unlabeled data as well as the extraction algorithm on this news data set. Its predictive performance is documented in Table 19. The performance is similar to the CRF classifier trained on the unlabeled data in Table 15 and achieves the highest predictive accuracy at the beginning of the address (class B).

Table 19. CRF model trained on the entire data sets

	Precision	Recall	F1-Score
B	0.80	0.91	0.85
I	0.48	0.93	0.63
E	0.85	0.77	0.81

After evaluating the performance of the classifier, then we applied the extraction algorithm to retrieve addresses based on the predicted labels. Table 20, Table 21 and Table 22 demonstrate the extracted results. We found 264 out of 325 addresses in the original news data set. Our method failed captured Water Street, Ivy Road, High Street, E Main Street, 221 E Main Street and Allied Street.

Table 20. Addresses extracted by CRF model

Address	Count	Address	Count
Swanson Drive	3	Preston Avenue	17
Fourth Street NW	9	Emmet Street	21
Second Street NW	3	Water Street	0
Monticello Road	13	Rugby Road	18
University Ave	6	Jefferson Park Avenue	20
Chesapeake Street	1	15th Street	9
Madison Avenue	5	Grady Avenue	5
Ivy Road	0	Gordon Avenue	1
High Street	0	Wertland Street	15
Elliewood Avenue	3	14th Street	10
Main Street	36	10th Street NW	2
5th Street	27	Sunset Avenue	1
Old Lynchburg Road	4	E Main St	0
4th Street	9		

Table 21. Close matches extracted by CRF model

Address	Block	Count
Elliewood Avenue		3
105 Lankford Ave	100	1
Madison Avenue		1
221 E Main Street	200	0
601 Preston Avenue	600	2
117 5th Street SE	100	2
319 E Main Street	300	3
209 Monticello Road	300	1

Table 22. Extracted addresses excluded by the police report

Address
900 South Street
Stone Creek Ln
16th Street
West Market Street
Solomon Rd
Allied street

Section 6: Conclusion and Future Work

6.1 Conclusion

This work was motivated by the under-report issue of sexual assault victims in order to find addresses from documents specifically news reports. Our research is built on the methodology of previous work on address extraction from web pages and builds CRF and XGBoost model with some additional features: minimum word distance to entity types and minimum word distance to "Sexual Assault" and related keywords. These extra features improved the model performance. From our experiments in both Washington Post and University Wire data sets, the CRF model achieved the most stable performance in both data sets. XGBoost model is the second best model but its main drawback is that it is less computationally efficient than the CRF model. Even though using PCA to preprocess the data can improve the performance in one of the data sets, it does not work well every time and also requires tuning number of principal components each time. In this case it might not be a good choice for semi-supervised learning which continuously add extra information from unlabeled data. We also tried a semi-supervised algorithm to improve the predictive performance on the University Wire data set. By using additional 900 unlabeled reports, we improved the weighted average F1-score of B, I, E classes from 0.72 to 0.79. Furthermore, we found 2 addresses of sex crime incidents found in the news documents which are not included in the crime report with the same time window.

6.2 Limitations and Future Work

Since there are no large labeled data sets like MUC-6 corpus for name entity recognition tasks accessible online, we have to manually label the data which limits our data size. We only focus

on two newspaper resources, Washington Post and University Wire (Cavalier Daily) only which might not represent the contents and styles in every newspaper. In this case, our model might not be generalized enough to predict address in every news report.

In the future, we should prepare a larger data which contains more diverse kinds of documents to help our model identify more generalized patterns. If the text descriptions of emergency record can be accessible, we can test our model on those records. Once we have a larger and more diverse data set, we can experiment with some other techniques such as word embeddings and recurrent neural networks and compare their performance with other models with hand-picked features. We should also try other more advanced and robust semi-supervised algorithms such as the semi-supervised CRF using generalized expectation criteria (Mann and McCallum, 2008).

References

[1] Ando, R., Zhang, T. (2005). A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. Journal of Machine Learning Research X.

[2] Bekele, R. & Menzel, W. (2005). A Bayesian Approach to Predict Performance of a Student (BAPPS): A Case with Ethiopian Students, *Artificial Intelligence and Applications*, Vienna, Austria, 189–194

[3] Breiding, M., Smith, S., Basile, K., Walters, M., Chen, J., & Merrick, M., (2014). Prevalence and characteristics of sexual violence, stalking, and intimate partner violence victimization-National intimate partner violence and sexual violence survey, United States, 2011.*MMRW* 2014; 63(No. SS-8).

[4] Brosi, M, (2013). Sorority Women's and Fraternity Men's Rape Myth Acceptance and Bystander Intervention Attitudes. *Journal of Student Affairs Research and Practice*.

[5] Chang, C., Li, S. (2010). Map Marker: Extraction of Postal Addresses and Associated Information for General Web Pages. Web Intelligence. pp.

[6] Chen, T., Guestrin, C. XGBoost: A Scalable Tree Boosting System. Retrieved from: http://arxiv.org/pdf/1603.02754v1.pdf

[7] Clougherty, E., Clougherty, J., Liu, X. and Brown, D. (2015). Spatial Temporal Analysis of Sexual Assaults in Charlottesville Area.

[8] Etzioni, et al. Unsupervised Named Entity Recognition from the Web: an Experimental study

[9] Goodchild, F. and Hill, L. L. INTRODUCTION TO DIGITAL GAZETTEER RESEARCH,

[10] Jolliffe, I.T. 2002. Principal Component Analysis. *Springer*, New York

[11] Nadeau, D. Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision. http://cogprints.org/5859/1/Thesis-David-Nadeau.pdf

[12] Nadeau, D., Sekine, (2007). A survey of name entity recognition and classification

[13] Lafferty, J., McCallum, A. and Pereira, F. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, pp

[14] Liao, W., Veeramachaneni, S. (2009). A simple semi-supervised algorithm for named entity recognition, in Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing, Boulder, Colorado, pp. 58–65.

[15] Okazaki, N. (2007). Crfsuite: a fast implementation of conditional random fields (crfs). Retrieved from: http://www.chokkan.org/software/crfsuite.

[16] Ratinov, L. and Roth, D. (2009). Design Challenges and Misconceptions in Named Entity Recognition

[17] Ringner, Markus. (2008). What is principle Components? Nature. pp. 303 – 304

[18] Sutton, C., McCallum, A. An Introduction to Conditional Random Fields of Relational Learning, pp 9 - 12

[19] Sharnagat, R. (2014). Name Entity Recognition: Literature Survey

[20] Smith, L. (2002). A Tutorial on Principal Components Analysis

[21] Sutton, C., McCallum, A. Introduction to Conditional Random Fields for Relational Learning

[22] Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. ACL.

[23] Yu. Z, (2007). High Accuracy Postal Address Extraction From Web Pages. Dalhousie University.

Appendix

Python code can be found at: https://github.com/wingsrc/newsReports