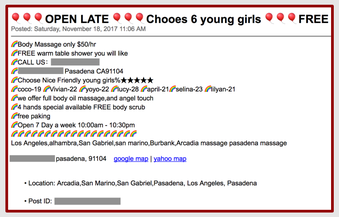
A Semi-Supervised Deep Learning Framework for the Automated Detection of Online Human Trafficking

Swetha Revanur and Keanu Spies
(srevanur, keanus) @ stanford.edu
CS221: Artificial Intelligence: Principles and Techniques

Human Trafficking

- Classified advertising has facilitated the vast majority of human trafficking in the United States. The National Center for Missing and Exploited Children found that a whopping 73% of child trafficking reports occur via Backpage.com.
- This is due to a lack of robust models to identify trafficking-related classified ads.



Objectives

- Design and develop an unsupervised filtering technique to identify posts likely to be related to human trafficking.
- Employ semi-supervised learning to build a more robust classifier.

Preprocessing

- A web crawler was built and deployed to recursively collect URLs for 2738 posts spanning 7 categories (Dating, MenSeekMen, MenSeekWomen, WomenSeekWomen, WomenSeekMen, Transgender, and Massage). Using batch processing, the corresponding post's ID, original text, title, date, location, phone number, and category were parsed from the HTML.
- Posts are informal and unstructured. To normalize and extract relevant textual data, preprocessing was conducted:

1. Strip HTML tags
2. Casefolding
3. Expand contractions
4. Strip phone numbers
5. Strip punctuations
6. Removal of one-character-long words
7. Emoji tokenization
8. Stop word removal

Phase I: Feature Engineering and Unsupervised Filtering

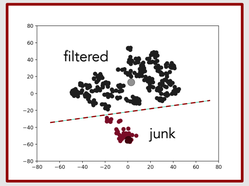
A. Feature Engineering

- Represent each post as a 19-dimensional binary feature vector:

Feature Group	Features
Language Pattern	third-person voice, first-person plural pronouns, Shannon entropy to assess text complexity, *n*-grams with TF-IDF (*n* = 3)
Keywords	[young, fresh, new, new in town, new arrival, open minded, petite, exotic, youthful, barely legal, virgin, tiny, incall, in call, new to the game, candy == 1]
Countries of Interest	[china, vietnam, korea, thailand, asian == 1]
Multiple Victims	[girls, women, men, boys, people, children, babes, dolls, masseuses == 1]
Victim Weight	[victim weight ≤110]
Spa Reference	[spa, massage == 1]
Presence of Emojis	[🌹, 🍒, 🌸, 💗, ✈️, 👑 == 1]

B. Unsupervised Filtering

- We obtain 1857 records from our dataset by filtering out samples that don't possess any of the binary features. This is our filtered dataset.
- To verify the fidelity of our filtering we apply the t-SNE transformation to create 2D projections of the filtered and junk feature vectors. We cluster these projections using *K*-means (*K* = 2).



Phase II: Semi-Supervised Classification

A. Expert Labeling

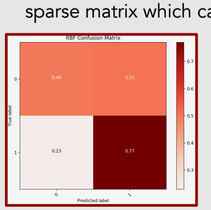
- Since we lack ground truth for our data, we rely on manual labels provided by experts.
- Of the 1857 records, 300 were labeled.

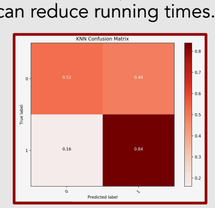
B. Baseline and Oracle

- The baseline is a majority algorithm for post classification. The majority algorithm classifies all the examples in the testing set as the majority class of the training set.
- The oracle is manual labeling of the entire filtered dataset.

D. Label Spreading

- Label spreading is a semi-supervised graph inference algorithm. Each unlabeled node inherits a label diffused from its similar labeled neighbors. The RBF kernel will produce a fully connected graph which is represented as a dense matrix. The KNN kernel will produce a sparse matrix which can reduce running times.



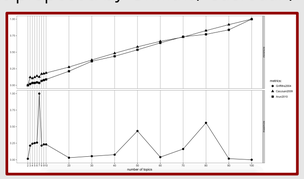


Semi-Supervised Classification Task Definition

- We have now reduced our task to a semi-supervised binary classification problem (semi-supervised due to the presence of both labeled and unlabeled data).

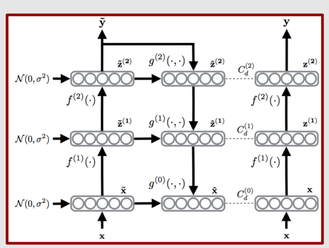
C. Feature Extraction with Topic Modeling

- Using Latent Dirichlet Allocation (LDA) topic modeling, we extract 7 of the most representative topics in the filtered dataset. LDA hyperparameters were tuned by evaluating the metrics proposed by Griffiths, Cao/Juan, and Arun.
- We generate a secondary 7-dimensional feature vector for each post from the document-topic distributions.



E. Deep Ladder Networks

- Ladder networks apply deep neural networks to simultaneously minimize the sum of supervised and unsupervised cost functions by backpropagation during training.



Phase II cont'd.

E. Deep Ladder Networks cont'd.

Algorithm 1 Calculation of the output and cost function of the Ladder network

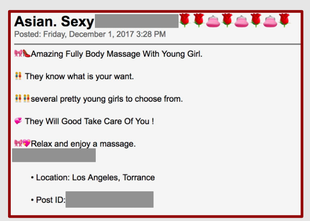
Require: $\mathbf{x}(n)$
Corrupted encoder and classifier
$\tilde{\mathbf{h}}^{(0)} \leftarrow \tilde{\mathbf{z}}^{(0)} \leftarrow \mathbf{x}(n) + \texttt{noise}$
for l = 1 to L **do**
$\tilde{\mathbf{z}}_{\text{pre}}^{(l)} \leftarrow \mathbf{W}^{(l)}\tilde{\mathbf{h}}^{(l-1)}$
$\tilde{\boldsymbol{\mu}}^{(l)} \leftarrow \texttt{batchmean}(\tilde{\mathbf{z}}_{\text{pre}}^{(l)})$
$\tilde{\boldsymbol{\sigma}}^{(l)} \leftarrow \texttt{batchstd}(\tilde{\mathbf{z}}_{\text{pre}}^{(l)})$
$\tilde{\mathbf{z}}^{(l)} \leftarrow \texttt{batchnorm}(\tilde{\mathbf{z}}_{\text{pre}}^{(l)}) + \texttt{noise}$
$\tilde{\mathbf{h}}^{(l)} \leftarrow \texttt{activation}(\boldsymbol{\gamma}^{(l)} \odot (\tilde{\mathbf{z}}^{(l)} + \boldsymbol{\beta}^{(l)}))$
end for
$P(\tilde{\mathbf{y}} \mid \mathbf{x}) \leftarrow \tilde{\mathbf{h}}^{(L)}$
Clean encoder (for denoising targets)
$\mathbf{h}^{(0)} \leftarrow \mathbf{z}^{(0)} \leftarrow \mathbf{x}(n)$
for l = 1 to L **do**
$\mathbf{z}^{(l)} \leftarrow \texttt{batchnorm}(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)})$
$\mathbf{h}^{(l)} \leftarrow \texttt{activation}(\boldsymbol{\gamma}^{(l)} \odot (\mathbf{z}^{(l)} + \boldsymbol{\beta}^{(l)}))$
end for
# Final classification:
$P(\mathbf{y} \mid \mathbf{x}) \leftarrow \mathbf{h}^{(L)}$
# Decoder and denoising
for l = L to 0 **do**
if l = L **then**
$\mathbf{u}^{(L)} \leftarrow \texttt{batchnorm}(\tilde{\mathbf{h}}^{(L)})$
else
$\mathbf{u}^{(l)} \leftarrow \texttt{batchnorm}(\mathbf{V}^{(l)}\hat{\mathbf{z}}^{(l+1)})$
end if
$\forall i : \hat{z}_i^{(l)} \leftarrow g(\tilde{z}_i^{(l)}, u_i^{(l)})$ # Eq. (1)
$\forall i : \hat{z}_{i,\text{BN}}^{(l)} \leftarrow \frac{\hat{z}_i^{(l)} - \tilde{\mu}_i^{(l)}}{\tilde{\sigma}_i^{(l)}}$
end for
# Cost function C for training:
$\text{C} \leftarrow 0$
if $t(n)$ **then**
$\text{C} \leftarrow -\log P(\tilde{\mathbf{y}} = t(n) \mid \mathbf{x})$
end if
$\text{C} \leftarrow \text{C} + \sum_{l=0}^{L} \lambda_l \left\| \mathbf{z}^{(l)} - \hat{\mathbf{z}}_{\text{BN}}^{(l)} \right\|^2$

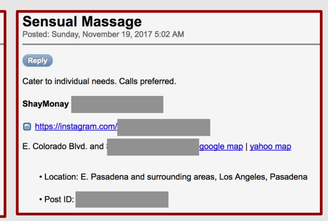
Results

- We report the preliminary accuracies of the the various approaches when trained on a subset of the filtered dataset:

Approach	Precision	Recall	Accuracy
Baseline Majority Algorithm	---	---	*56.667%*
Oracle	100%	100%	100%
Label Spreading (RBF)	64%	64%	63%
Label Spreading (KNN)	70%	68%	67%
Deep Ladder Networks	---	---	*80%*

- We observe that our TensorFlow implementation of Deep Ladder Networks achieves the highest accuracy. On the left, is a Backpage post correctly classified as human trafficking. On the right is a post correctly classified as a consensual activity.





Selected References

- Alvari et al., A Non-Parametric Learning Approach to Identify Online Human Trafficking
- Rasmus et al., Semi-Supervised Learning with Ladder Networks