Analyze Sex Trafficking Data using Neural Network Approach

Ruyuan Zuo
Department of Data Science and Computer Science
Case Western Reserve University
Cleveland, OH, United States
rxz250 [at] case [dot] edu

ABSTRACT

There are about 600,000 to 800,000 human trafficking cases each year, and over 50% of these victims are victims of sex trafficking [1]. Traditionally, sociology scholars had concluded few factor that mediating the trafficking victim to sex trafficking. In this project, I will be looking into the mediating effect of age, gender, recruiter relationship and country of exploitation as factors to sex trafficking. We will be using a four-layer dense neural network approaches to construct a deep network to classify cases that are positive for sex trafficking among learning from a public available dataset from Counter Trafficking Data Collaborative (CTDC) [2]. I will also explore the effect of words that contain information on the above factors through a textural understanding network using word embedding and Long Short Term Memory (LSTM) layer. The result finds a successful classification of the mediating factors, as well as demonstrated the potential of LSTM with wording embedding on understanding textural information related to sex trafficking.

1. Background

Human trafficking is a hateful crime and considered as the modern-day slavery. This project will be focusing on sex trafficking, which is one of the most profitable industry among all sectors of human trafficking. According to Zhang and Yen, sex trafficking produces $7 to $12 billion per year [3, 4].

This project will construct a deep neural network designed for looking into indicating factors for sex trafficking among all human trafficking victims, as well as a textural understanding of sex trafficking related information in text.

Acquiring from public sources, the global dataset contains over 488802 raw datasets that contains information like age, sex, country of origin and the exploiting type. Of these data, we clean the dataset with filtering out the incomplete information, convert the categorical data into discrete data, and then divide the data sets into train and test. Our result with four hidden layer and drop outs neural network can achieve an 97% accuracy with loss around 0.08. Suggesting that the network has successfully find a weights that can successfully classify the class of victim as being sex exploited (1= sex trafficking) or not (0 = human trafficking).

The second part of the project focuses on textural analysis, as suggested by a research article by Tang et al [5]. They suggested in their research that for modern counter-human trafficking operations, a multimodal structured needs to be employed with the complexity of the forms of communications [5]. As an example, a trafficker will post their advertisement for services online with both text and images. As an inspiration, I would want to know how would a network structure like LSTM with word embedding would have a strong performance with the recurrent structure for analyzing text. Due to the security of the sensitive data, there are seemed to be a lack of information on the publically available datasets for sex trafficking. For this reason, for the second part, I used an .nltk package with context freed grammar to generate random sentences, and used word embedding to vectorize the sentences, then used LSTM layer to find classifications for the sentences where I embedding some critical information that part one had analyzed, like age, gender, country and relationship. The result of 3185 manually generated data with labels on sex, gender and citizenship with recruiter relationship can achieve a best result of 0.97 in one epoch.

However, this result is of flaw because the way that the data information is generated by a certain grammar, it is very possible that the network has remembered the data, as well as remembered the key words.

As a limitation of lack of real data, and the lack of the expertise of humans trafficking, I found it is not possible for me to label the semantics of the sentence by myself for random sentences. However, as the finding of this project has suggested, if we were to have a set of data that is properly labeled by a human trafficking experts, using word embedding and a LSTM layer can effective find the key words that is in the sentence and find the convergence very fast.

2. Methods

2.1 Part One: Deep network analysis on structured data

2.1.1 Describe the data set and pre-processing steps

The dataset we get is from The Counter Trafficking Data Collaborative (CTDC). The original data set has 11 inputs and a label with forms of exploitation. I especially looked at the variable "isSexualExploit" and set this as the label class. As research on sex trafficking has suggested that age, gender, recruiter relationship and country of exploitation has relationships with the manipulations used for keeping victims of sex trafficking captive. As a result, I will particularly look into these few factors when designing the network.

After performing data cleaning entries and then covert categorical data to discrete values see (appendix I, code_display), the Table 1 below shows an example of the cleaned data table. The final N = 14255 with 11992 positive cases and 2263 negative case.

In order for the network to be unbiased toward both the positive class and the negative class, I performed the following analysis with neural network on random selected 2263 positive cases and 2263 negative cases to reach a balanced dataset. In further analysis, I find that there is no significant different with using a skewed distribution for learning compare to an even distribution. However, for the following analysis, I will keep an even distribution of classes.

	gender	ageBroad	RecruiterRelationship	CountryOfExploitation	isSexualExploit
9840	2	7	4	4	0
9842	2	7	4	4	0
9843	2	7	4	4	0
9844	2	7	4	4	0
9845	2	7	4	4	0

Figure 1: example dataset for Part One

2.1.2 Network structure

The input of the network is a four by one vector, the four hidden layers with two layers of dropouts at rate 0.25 is show below.

```
Model: "functional_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
int (InputLayer)             [(None, 4)]               0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                160
_________________________________________________________________
dense_2 (Dense)              (None, 16)                528
_________________________________________________________________
dropout (Dropout)            (None, 16)                0
_________________________________________________________________
dense_3 (Dense)              (None, 32)                544
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_4 (Dense)              (None, 64)                2112
_________________________________________________________________
predictions (Dense)          (None, 2)                 130
=================================================================
Total params: 3,474
Trainable params: 3,474
Non-trainable params: 0
_________________________________________________________________
```

Figure 2: Neural network design

The choice of the layers was considering that the inputs for the dataset are limited to four inputs, so the network will not learn as much if the interneurons are two large. However, consider the input range of the CountryOfExploitation has 20 different classes. I choose the first layer to be at 32 interneurons with the second layer at 16 interneurons. The design for the choosing 3rd and 4th layer to be at 32 to 64 interneurons is an experimental choice when testing the performance with 16, 32, and 64 to come up with the best choice of 32+64 with dropout rate 0.25 to remove uninformative neurons.

The design choice for a 4-layer structure compared with 3 seemed to have better overall performance. With 3 dense layer at 32, 16, 64, the network converges faster at 4 layers compare to three. While reaching the same low error rate. Under this comparison, I choose 4 layers in order to maintain the performance for a larger dataset.

The design choice for implementing only dense layers is in consideration for the structured data type does not have features for using recurrent our convolution network, since the input size is small and the data is structured.

The choice of activation is RELU is because we don't need negative inputs and these negative data will have no realistic meaning in the context for any of the inputs parameters.

2.2 Part Two: LSTM textural analysis

The second step is for looking into textural understanding using word embedding and long short term memory (LSTM).

2.2.1 Describe the data set and pre-processing steps

Since there is no open available dataset for annotated sex trafficking dataset, I decided to create a dataset that contains annotated information on the four criteria that I checked in part one.

For the zero labels, my original choice is to have the a define grammar with nature language processing toolkit (nltk) to parse through the grammar and then generate a random sentence. However, since I am suing the same technique for generating the annotated dataset, I employed a dataset developed by the Naval Postgraduate school. The specific dataset I will be using is called "10-19-20s_706posts.xml", which contains 3185 sentence entries from online forms.

	class	txt
0	Statement	now im left with this gay name
4	Statement	ah well
11	Statement	26/ m/ ky women that are nice please pm me
14	Statement	there ya go 10-19-20sUser7
20	Statement	i'll thunder clap your ass.

Figure 3: example dataset from Naval Postgraduate Academy (NPS)

These entries are labeled negative, meaning that the wording does not contain annotated information on gender, age, or countries information.

Then for generating the selected sentence with a defined grammar the following grammar is used.

1. S -> NP VP
2. PP -> P NP
3. NP -> Det N | Det N PP | 'I'
4. VP -> V NP | VP PP
5. Det -> 'an' | 'my' | 'young'
6. N -> 'girl ' | 'women'| 'chick'| 'China'
7. V -> 'shot' | 'want' |'service' | 'sex'
8. P -> 'in'

Pseudocode 1: context free grammar example

The meaning for S is stand for sentence, the definition above stated that a sentence S can be composed of NP (noun phrase) or VP (noun phrase). A noun phrase NP composed of a determinant (Det), a noun (N), or a determinant (Det), a noun (N) and a proposition (PP), etc. From line 5 to 8 defines the noun word that can be used for the generation of the sentences. In the noun category, there is word choice that segmented on gender ("women"), nationality ("China"), and age ("young").

Using the context free grammar define above, we generate same number of class positives for the training set, and labeled as one. It is true that sentence generation method is simple, and because of the lack of true labeled data, this method is only aim for demonstration of the power of RNN structure like LSTM with wording embedding can learn quickly the weighted words, in this case sensitive words like gender and age. With a segmented data, it would offer a possible solution to analyze the semantics.

For the next step, the method tokenizer is called for converting sentence to word tokens, which is a vector of integers that represent each word.

```
[[10, 16, 20, 11, 17],
 [11, 16, 19, 1, 11, 16],
 [10, 16, 21],
 [10, 16, 20, 10, 17],
 [11, 16, 19, 1, 11, 17],
 [11, 16, 21],
 [10, 17, 19, 1, 10, 18],
 [10, 17, 19, 22, 11, 17],
 [11, 16, 20, 10, 18],
 [11, 18, 20, 11, 16],
 [10, 16, 19, 22, 11, 16],
 [11, 18, 21],
 [10, 17, 21],
 [11, 18, 19, 22, 11, 18],
 [10, 16, 19, 1, 10, 17],
 [11, 17, 20, 11, 18],
 [11, 18, 19, 1, 10, 17],
 [10, 17, 20, 10, 17],
 [10, 16, 19, 22, 11, 17],
```

Figure 4: tokenized sentences

Then, a padding function to turn into a matrices of the frequency of the word with respect to each sentence.

	0	1	2	3	4	5	6	7	8	9	...	57	58	59	60	61	62	63	64	65	6
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	10	16	20	11	1
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	11	16	19	1	11	1
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	10	16	2
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	10	16	20	10	1
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	11	16	19	1	11	1

Figure 4: padded sequences in matrices form

This matrix is split into train set and testing set, and inputted to the neural network.

2.2.2 Network design

Since the dataset is embedding with the tokenizer (with max length of 2500), the first layer is an embedding layer with input size of the max embedding length 2500 and output size of 128. Then this normalized vector is input into a LSTM layer to perform a classification of whether the input sentence contains information on sex trafficking (label = 1) or does not (label = 0).

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
int (InputLayer)             [(None, 67)]              0
_________________________________________________________________
embedding (Embedding)        (None, 67, 128)           320000
_________________________________________________________________
lstm (LSTM)                  (None, 200)               263200
_________________________________________________________________
predictions (Dense)          (None, 2)                 402
=================================================================
Total params: 583,602
Trainable params: 583,602
Non-trainable params: 0
_________________________________________________________________
```

Figure 5: Network Structure for Part Two

3. Results

3.1 Part One result

The result sheet with varying parameters

num of layers	activation function	nodes per layer	dropout rate	last epoch error rate %
4	relu	(32,16,32,64)	0.25	0.9711
4	relu	(32,16,32,64)	0.5	0.9718
4	sigmoid	(32,16,32,64)	0.25	0.9705
3	relu	(32,16,64)	no	0.9697
3	relu	(32,16,64)	0.25	0.9708
3	sigmoid	(32,16,64)	0.25	0.9711

Table 1: result sheet for part one

The best result achieved overall is of accuracy on the training set of 0.97%.

3.2 Part Two result

The best result with 10 epochs can result in 100% recognition. I think the reason why is because the manually generated sentence, rather the real semantics, or the real meaning of the sentence were used to train the network. The network remembered the keyword that were put on emphasis on. Even though the approach in this case is reverted, however, the performance of LSTM is very strong, within three epochs, the network can have achieved an almost exact recognition of the key word with accuracy = 100%. The time it takes for the network to find converge in each epoch is very small as well. As a result, I think word embedding with the LSTM is a very power tool for nature language processing, as exampled by this experiment, but this specific network structure is not for use in real settings because the ineffective training data.

4. Discussion

The experiment can consider as successful, by input parameters like age, gender, recruiter relationships and country of exploitation, the dense network is able to recognize a weight that can classify either a victim is of risk to sexual exploitation. For future work, it might be helpful to consider more factors so that the training input size is not too small compared to the interneurons,

For the second part, as noted in the result section, the dataset need semantically labeled sex trafficking data by human experts to be able to perform more power classification task. Yet, this experiment still shows the power of LSTM performance. The future work is to find data that are being labeled by human expertise, if possible, and apply the word embedding with LSTM layer again.

References

[1] C. Giordano, "Reinventing the Wheel: Returning Sex Trafficking Discourse to Its Basic Human Rights Origins Note," *Suffolk Transnational Law Review,* vol. 37, no. 2, pp. 347-375, 2014.

[2] "The Counter Trafficking Data Collaborative (CTDC)," [Online]. Available: https://www.ctdatacollaborative.org/. [Accessed 4 December 2020].

[3] Z. Sheldon, "Beyond the 'Natasha' story – a review and critique of current research on sex trafficking," *Global Crime,* vol. 10, no. 3, pp. 178-195, 2009.

[4] I. Yen, "Of Vice and Men: A New Approach to Eradicating Sex Trafficking by Reducing Male Demand through Educational Programs and Abolitionist Legislation," *J. Crim. L. & Criminology,* vol. 98, no. 2, pp. 653-686, 2008.

[5] E. Tong, A. Zadeh, C. Jones, et al., "Combating Human Trafficking with Deep Multimodal Models".

[6] Stepthom, "How to use NLTK to generate sentences from an induced grammar?," [Online]. Available: https://stackoverflow.com/questions/15009656/. [Accessed 4 December 2020].

[7] Python. Inc, "Categorizing and Tagging Words," Natural Language Toolkit - NLTK 3.5 documentation.," [Online]. Available: http://www.nltk.org/book_1ed/ch05.html.. [Accessed 4 December 2020].

[8] "Understanding LSTM and its quick implementation in keras for sentiment analysis," Medium, 03 March 2018. [Online]. Available: https://towardsdatascience.com/understanding-lstm-and-its-quick-implementation-in-keras-for-sentiment-analysis-af410fd85b47. [Accessed 4 December 2020].