Combating Human Trafficking via Automatic OSINT Collection, Validation and Fusion

Bibek Upadhayay, Zeeshan Ahmed M Lodhia, Vahid Behzadan

[1]SAIL Lab
University of New Haven
West Haven, Connecticut 06516
bupad1@unh.newhaven.edu, zlodh1@unh.newhaven.edu ,vbehzadan@newhaven.edu

Abstract

A major challenge in combating global human trafficking is the availability of actionable intelligence about trafficking events and operations. The lack of timely and structured data remains a significant bottleneck in the monitoring and mitigation of human trafficking. In this collaborative work with Love Justice International[1], we aim to address this issue by developing an automated pipeline based on recent advances in natural language processing and machine learning to streamline the curation, analysis, and extraction of actionable intelligence from multi-sourced news media as Open-Sources of Intelligence (OSINT). In our solution, we utilize and enhance the BERT Question Answering language model for information extraction from the unstructured text of the news. Furthermore, we develop algorithms for measuring the relevance and novelty of curated news articles to reduce the computation cost and redundant processing. Moreover, we evaluate the proposed pipeline on a dataset of annotated news articles containing actionable intelligence about victims and perpetrators of human trafficking.

Introduction

Human trafficking has been a widespread issue, affecting thousands of people globally. In 2017 alone, the International Labour Organization (ILO) estimated the number of trafficking victims at 40.3 million worldwide. [2]. Furthermore, the scale of human trafficking operations has been growing rapidly in recent years, and the Polaris Project estimated a 20% increase in such operations in 2018 – 2019. The problem of human trafficking affects men, women, and children around the world. The victims of human trafficking face different kinds of exploitation such as sexual exploitation, forced labor, slavery, children are used as child soldiers and child brides and for begging. The ILO also reports [3] that forced labor and sexual exploitation generates 150 billion U.S. dollars annually, which implies that human trafficking is a global and enormous problem and will keep on increasing until and unless it is tackled.

The International Government Organizations(IGOs), Non-Governmental Organizations (NGOs), and United Nations have come together to combat human trafficking at local, national, and international levels. The government and the policymakers have strict rules and policies regarding human trafficking by prosecuting the traffickers and providing protection and rehabilitation to the survivor. This resulted in the trafficking industry being more secretive and hard to recognize. There are different challenges that exist with the identification of trafficking and combating it. The traffickers try to avoid detection and keep on adapting by changing their patterns which includes changing the advertising websites, phone number, mode of transportation, and so on. The traffickers also limit the societal interfaces of victims by different means such as threats, violence, and drug abuse. This secretiveness leads to the lack of comprehensive data which is another challenge.

Comprehensive data plays an important role in combating human trafficking. The data and its analysis would provide the entities involved in counter-trafficking with insights on the trends and scale of trafficking operations, as well as information about methods used by traffickers. This can include how victims are lured, captured, and exploited, thereby enabling the development of new policies and strategies for combating human trafficking.

Government entities and NGOs work at the community level to collect data from multiple sources such as calls, emails, web applications that are unstructured and heterogeneous in nature. However, the intelligence collected by these stakeholders is generally constrained to certain localities, fragmented and disconnected, since such organizations collect data for their internal purposes and there is reluctance in sharing these data. This reluctance in sharing the data has resulted from questions of privacy of data, the credit of data, and also the purpose of uses of data. As a result, the lack of comprehensive data makes it difficult for analysis and research.

A promising solution to the problem of obtaining insights and intelligence at a global scale is to make use of available data on the web such as social media, different local newspa-

[1] https://www.lovejustice.ngo/

[2] http://www.ilo.org/global/publications/books/WCMS_575479/lang--en/index.htm

[3] https://www.ilo.org/global/publications/ilo-bookstore/order-online/books/WCMS_243391/lang--en/index.htm

pers, forums, blogs, and other related websites. Particularly, news articles from news media can provide information on human trafficking incidents at both local and international levels. Such news articles may contain information such as biographical details of victims and suspects, as well as operational information such as how the victims were captured and lured, what made the victims migrate, mode of transportation, whether the suspects were arrested or not, and the means of exploitation. However, a significant bottleneck in this process is the limited bandwidth of human investigators in curating and analyzing these sources to produce timely and actionable intelligence from these sources at a global scale.

There are initiatives that have been taken for creating and sharing them on a global level. Counter-Trafficking Data Collaborative[4] is a global dataset collected from organizations all over the world. However, the dataset is not timely and does not provide information on victims and suspects. International Centre for Missing Exploited Children [5] (ICMEC) maintains the global missing children's dataset. The ICMEC data only contains the information on children but not on adults and the data is collected from a limited number of countries only. There is a lack of readily, timely, and more comprehensive updated data. Hence, the current body of work in this domain includes efforts towards collecting data on human trafficking. (Brewster, Ingle, and Rankin 2014) proposes a framework for crawling open-source data for information and indicators of human trafficking for facilitating detection and analysis of indicators. (Hundman et al. 2017) presents an end-to-end tracking and detection pipeline for multi-modal sources of data via crawling advertisements for sexual services on the web. (Tong et al. 2017) curated a dataset of 10,000 annotated advertisements. In general, previous efforts in this domain are mostly focused on identifying the indicators of human trafficking from specific contexts, and thus the issue of intelligence extraction from unstructured and generic news articles is yet to be addressed. Moreover, there remain significant gaps in meeting the need for a fully automated process of information extraction in this domain.

In this collaborative work with Love Justice International, we aim to address the above-mentioned issues by leveraging natural language processing and unsupervised machine learning methods to create a fully automated pipeline for curation, analysis, and extraction of intelligence on human trafficking events. We developed an automated OSINT collector that curates raw multi-source intelligence from the news media and utilizes the BERT Question Answering language model for information extraction to produce actionable intelligence. Love Justice International uses actionable intelligence to monitor and identify trends and patterns in trafficking operations for mitigation and policy purposes and provides supports to victims of human trafficking. We also develop algorithms for measuring document similarity and relevance to further improve the computing cost and accuracy of the proposed pipeline. Accordingly, the main contributions of this paper are as follows:

- we develop a fully-automated pipeline for OSINT extraction and analysis of unstructured text collected from multi-sourced news media,
- we develop algorithms for news similarity matching and relevance checking to filter redundant data, thus reducing the computation cost,
- we leverage the BERT QA model to extract actionable intelligence, and improve its performance via the use of third-party APIs for better extraction and categorization of intelligence.

Background

OSINT collection in Human Trafficking Open Source Intelligence (OSINT) is the intelligence that is extracted and derived from publicly available sources. The sources can be websites, news websites, blogs, government and non-government organizations reports, academic papers, Wikipedia, social media content. OSINT is not only popular among academicians but as well as among many law enforcement agencies. The advantage of analyzing big digital forensic data is that it might contain case-related information contained within disparate data sources (Quick and Choo 2018). The actionable intelligence of the case will help decision-makers to independently and impartially inform decision (Gibson 2004). OSINT collection has been the key step in combating human trafficking using machine learning. In the DARPA MEMEX [6] program, researchers collected hundreds of millions of online sex advertisements which has been the resource for many researchers on finding the indicators of human trafficking in online advertisements. Similarly, the Trafficking-10K dataset (Tong et al. 2017) is also an example of OSINT collection for the human trafficking dataset.

BERT Language Model Bidirectional Encoder Representations from Transformer (BERT) (Devlin et al. 2018) is a state-of-the-art word representation model that uses a transformer to learn the contextual relationships between the given text in a bidirectional manner (i.e., in both left-to-right and right-to-left). The bidirectionally of BERT has made it stand in many NLP tasks, as it improves fine-tuning-based approaches to token level tasks. BERT uses two strategies in training, the first one is Masked Language Model (MLM) and the second one is Next Sentence Prediction (NSP). In MLM, the model randomly masks some of the tokens from the input text and then the model tries to predict that masked vocabulary id of the word. It does so by looking into the context in both directions. As for Next Sentence Prediction, the pairs of sentences are given input to the model and it tries to predict if the given second sentence is the continuation or the next subsequent sentence to the first one or not. The model uses both MLM and NSP during training to minimize the loss. BERT can be modified to perform many downstream tasks such as classification, question answering, summarization, sentence prediction, and so on.

[4] https://www.ctdatacollaborative.org/

[5] https://www.icmec.org/

[6] https://www.darpa.mil/program/memex

BERT for Question answering BERT for Question Answering transformers is BERT Model. BERT can be modified to perform many downstream tasks by feeding the task-specific inputs and outputs. For question answering tasks like SQuAD(Rajpurkar et al. 2016) it has a span classification on top of the hidden state output to compute the span start logits and span end logits. The transformers from the HuggingFace[7] are pre-trained on SQuAD(Rajpurkar et al. 2016) . The BERT QA model takes two inputs, a question and a reference text containing an answer. The question and the reference text are separated by the [SEP] token. BERT also makes use of 'Segment Embeddings' to differentiate the question from the reference text, and these embeddings are learned by transformers which it adds to the token embeddings before feeding into the input layer. The output of the BERT Question Answering model is simply highlighting the span of the reference text from start token to end token. This is achieved by model finding the start token and end token. The final embedding of all tokens in the text is fed into the start token classifier. The start token classifier applies a single set of weights to every word and takes a dot product between the start weights and the output embedding which is later applied with softmax activation. The final probability distribution from softmax activation gives the word with the highest probability distribution which is then selected as the start token. The same process is repeated for the end token with a separate weight vector. BERT for Question Answering is the BERT-Large model that has 24-layers and an embedding size of 1024 with a total of 340 parameters.

Related Work

The breakthrough in machine learning, computer vision, and natural processing has made a significant impact on developing tools for combating human trafficking. (Tong et al. 2017) designed a deep multimodal model called Human Trafficking Deep Network (HTDN) that detects the human trafficking ads in the Trafficking-10K dataset. Trafficking-10K dataset is the contribution from authors which is the first annotated dataset for detection of human trafficking. (Zhu, Li, and Jones 2019) developed and extended HTDN the previous tool that learns language structures in adult service ads to detect human trafficking ads which can also automatically identify keywords in suspected sex trafficking ads even when the ads were written with obfuscation techniques. The authors achieved the F1 score of 0.696 and applied their tool for the detection of human trafficking organizations. FlagIt (Kejriwal et al. 2017) is another tool used for indicator mining that is widely used by more than 200 law enforcement agencies that look for five indicators of human trafficking. FlagIt stands for Flexible and adaptive generation of Indicators from text, is based on supervised machine learning and unsupervised text embedding, and semi-supervised heuristic re-labeling for discovering indicators in the web that can be used for lead generation and lead investigation. (Mensikova and Mattmann 2017) trained binary and multi-class sentiment model for identifying human trafficking using escort review data crawled from the open web. As we can observe that the most of the works are related to identifying human trafficking or finding indicators of human trafficking. Our work focuses mostly on developing automated pipelines for OSINT extraction for preventing human trafficking at the community level.

(Shi and Lin 2019) demonstrate the use of BERT-base model for relation extraction and semantic role labeling. The task of relation extraction is to predict if there is any relation between two entities or not. The author shows that the pre-trained BERT can be used for relation extraction without relying on lexical or syntactic features. (Yang et al. 2019) used BERT with an open-source Anserni retrieval toolkit that identifies answers from a large corpus of Wikipedia articles in an end-to-end fashion. (Yang, Zhang, and Lin 2019) used BERT for improving the ad hoc document retrieval. The authors also address the issue of processing longer documents than the length of input BERT was designed to handle. Authors feed each sentence individually to the model and then aggregate the sentence scores to produce document scores. These related works are similar to the objective of our research project so to extract information from the documents we made use of the BERT language model.

Methodology

The proposed solution is comprised of 2 main components, namely the news crawler and the information extractor. This section provides the technical details of both components, as well as the flow of the corresponding processes.

News Crawler

The News Crawler is an application designed to crawl the news from the web. The objective of this component in our solution is to search the web for news articles that are related to human trafficking and contain actionable intelligence about victims or suspects. The procedural flow of the news crawler is illustrated in Figure 1. The news crawler searches for the news using the Google News API, and stores the news in the MongoDB database. Curated articles are then passed to the news similarity checker, where similar and duplicate articles are removed. The remaining articles are then passed to the news relevance checker, which filters out the non-relevant articles and updates the MongoDB database with the remaining documents.

The crawler uses search phrases related to human trafficking and extracts the news articles returned by the Google News API. In our project, it is of importance to collect news articles that are related to various human trafficking topics such as child laboring, sex trafficking, human trafficking, etc. Thus the challenge arises as to what should be the search phrases to yield and extract more relevant news articles to meet this objective.

KeyBERT and the Search Phrases We made use of the KeyBERT (Grootendorst 2020) model which is the modification of BERT that gives the keywords as an output for the given text. We feed the titles of articles from a pre-annotated dataset of 366 articles related to human trafficking to the model and retrieved a set of descriptive keywords as the output. We only selected the keywords that occurred

[7] www.huggingface.co

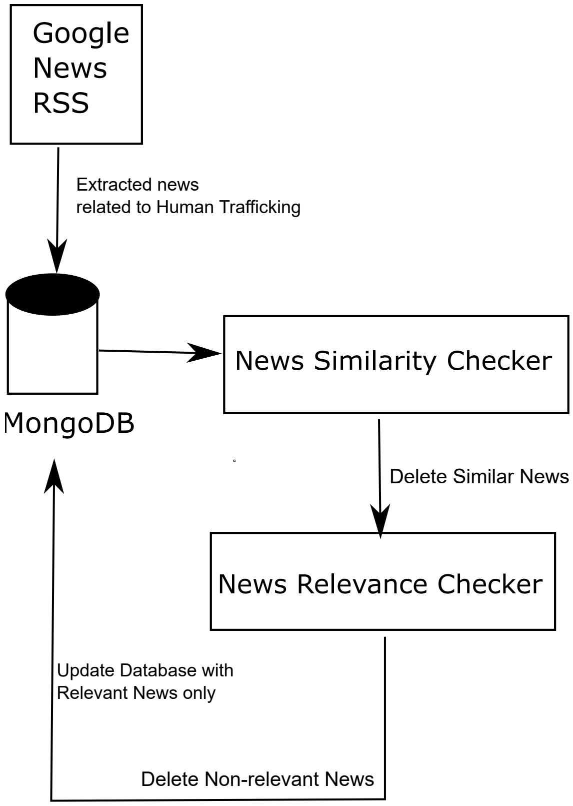

Figure 1: News Crawler Process Flow

with high frequencies. We also included the name of countries that are of interest to our sponsor (i.e., Love Justice International) in the keyword list. Accordingly, we formed the search phrases by combining high-frequency keywords and the name of countries in which Love Justice operates. Examples of search keywords include: Human Trafficking, Child Trafficking, Labor Trafficking, Human Trafficking in Nepal, Child Trafficking Nepal, Labor Trafficking Nepal, Human Trafficking India, Traffickers Arrested Nepal, Traffickers Arrested India and so on. The list of countries and search keywords are provided in GitHub.

Use of Google News RSS for news crawling: We used two methods for new searching. The first method was to search news using the Google search engine. In this method, the results generated from Google were from news websites as well as other websites. During the manual evaluation of crawled news, much of the news was found to be irrelevant. This motivated us to explore the second method, to use the Google News RSS for crawling news. Google News provides aggregated results from news websites around the world. The manual evaluation of crawled news was highly relevant to human trafficking. Hence, the second method was implemented in the crawler.

News Similarity Checker After the crawling and extraction of the news, the news is saved in the database. The curated news is based on the combination of search phrases, and there are high chances of crawling the same news with the same link because the combination of search phrases shares a similar semantic to Google News Engine. There are also cases where the same case of human trafficking is shared or published by different news media. There also might be the case where subsidiary companies of media covering the same news with different titles but the case and context being same. All these cases resulted in the large numbers of crawled news that are similar i.e. the same human trafficking case covered in multiple news articles.

The solution to this problem is the News Similarity Checker that checks for similar news in daily crawled news and flagged them as similar. The program is based on a scoring mechanism. The observation of the human trafficking dataset helps to define a hypothesis that two news articles that are similar and cover the same human trafficking case will have similar titles and share the same-named entities in the news content. The formulation of the program based on this hypothesis has a process where each article is passed through the program and check against other articles to determine whether it is similar or not. The articles are passed through two steps and in each step a score is provided to the article, if the article's total scores pass that threshold score, then it is flagged as similar. In the first step, the titles of the news are compared with cosine similarity. The cosine score is then added to the total score of the article. In the second step, the program compares the number of named entities in both articles. The named entities for both the articles are extracted using IBM Watson API and then compared. The scoring mechanism is set up for each kind of entity's match. For example, if one person's name is matched then a score of 25 is added to the total score, if 2 person's names are matched the score of 35 is added to the total score. Similar scoring is provided based on the location, organization, and quantity of entities matched between the two articles. Finally, the total score is compared with the threshold value and if the score is greater than the threshold the article is flagged as similar.

News Relevance Checker The objective of the news crawler was to collect news articles that contain information on human trafficking and have a high probability of containing victims' and suspects' details. There were two additional challenges faced, the first challenge was that the articles were related to human trafficking but lacking the information on suspects' and victims' details. The second challenge was that the articles did contain the search keywords combinations or few keywords on human trafficking, for example, a famous person being a victim but the articles overall context was not related to human trafficking. The latter problem also gave rise to over-polluted news since news sources cover famous persons at high frequency. In summary, the problem was to find the relevant articles. The relevant article finding process can be defined as finding the news articles related to human trafficking and contain information on victims or suspects.

In order to overcome this challenge, we implemented a news relevance checker that will flag the non-relevant news articles. News relevance checker algorithm accounts in two steps checking in order to determine the relevance of articles to meet the objective of news crawler. The first step checks for the presence of the relevant keyword in articles and the second step determines whether the person named entities presented in the articles are a famous person or not. The presence of relevant keywords is simply checking whether the set of keywords are presented in the articles or not. The keywords set is derived based on the keywords that are used in a set of questions fed to the Information Extraction Model. If none of the keywords in the set are presented in the articles then the article is flagged as non-relevant and will not process further. On the other hand, if the article is flagged as relevant it is then passed for the second step. The articles containing famous person name in them often almost turn out to be non-relevant articles. The question of solving how to determine whether the person mentioned in an article is famous or not is solved by using the Wikipedia API. The hypothesis used here is that if the person is famous then that person's Wikipedia page must exist. The algorithm sends each person's name extracted using IBM to the Wikipedia API to determine whether the Wikipedia page exists or not. If the Wikipedia page has existed then the article is flagged as non-relevant and is not processed further.

Information Extractor

The news articles are passed through the information extractor to retrieve important information about victims and suspects. The objective of this component is to retrieve the following information: victims' names, age, gender, nationality, address, education, occupation, and guardian, as well as suspects' names, gender, age, nationality, and address. In addition to victims' and suspects' details, the extractor also looks for the following information: the promise made to victims by suspects, victims' purpose of travel, and the form of exploitation victims faced (e.g., labor, sex, etc.). The information extractor is comprised of a Text Filter and the BERT QA Language Model. After passing through the news similarity checker and relevance checker, news articles are passed to the text filter and then to the language model for the information extraction. The block diagram of the information extractor is illustrated in Figure 2.

Text Filter The text filter pre-processes the news articles before passing them to the model. The pre-processing stage includes the removal of unwanted HTML tags, links, non-alphabetical ASCII characters, and emojis.

The information extraction process is done using the BERT Question Answering Model. The question and the reference text containing the answer are feed into the model and the model provides the answer and the score. For instance, to find the victim in an article, we pass the question 'Who was the victim?' and the news article as a reference text to the model. The model class is defined in such a way that it only provides the answer if the confidence score produced by the model is greater than a user-defined threshold (e.g., 90%). This threshold can be tuned by the user as a configuration parameter of our solution.

Questions and Dynamic Questions In order to find each information, the model needs to be asked a specific question. The age and gender of the victim can be identified by asking "What was the age of the victim?" and "What was the gender of the victim?". However, it might not always be the case where an article explicitly mentioned the keyword 'victim'. The article might talk about an actual victim with the use of similar keywords such as survivor, abductee, or rescued person. In such cases the answer yielded by asking the question 'Who was the victim?' or 'What was the age of victim?' might not be correct or the answer might have a score below 0.90. The solution to overcome this problem is to ask different questions to get the same answer. The notion of representing victims or suspects in articles differs vastly from authors of articles to media publishing the article. Utilizing KeyBERT and considering the top keywords in the articles helps to formulate sets of questions that can be passed to the model to extract each information. For instance, to find the name of victims, the following questions are composed: *Who was the victim? Who was kidnapped?, Who was rescued? Who was trafficked? Who was the survivor?*. The name of the suspect can be found by asking questions such as: *Who was arrested? Who was the kidnapper? Who was the suspect? Who was the abductor?*.

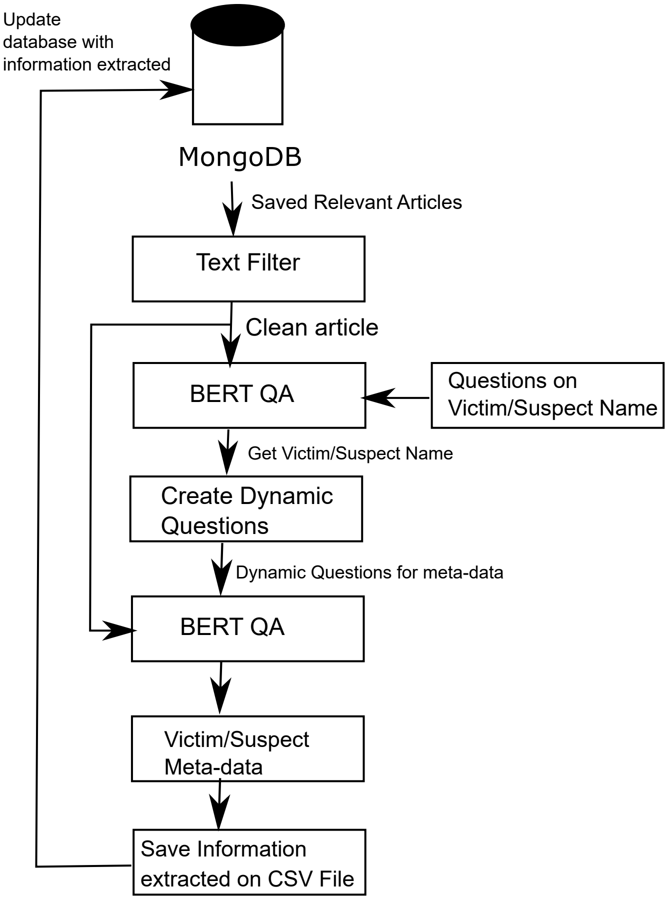

Figure 2: Information Extractor Process Flow

For extracting each information its corresponding sets of questions are passed to the model, which yields answers and scores for each question. Then our algorithm combines the answers from that set of questions and selects the answer with the highest score for that respective information.

Similar questions can be composed to determine other information of interest for the victims and suspects. For instance, to extract the age of a victim, the set of questions can be formed as *'What was the age of the victim?', 'What was the age of survivor?', 'What was the age of abductee?'*. This approach of extracting relevant information results in asking the same amount of questions from the model as compared to finding victims and suspects. While the model performs well in extracting the information, the generic structure of the aforementioned questions gives rise to high computing time and cost for each article. Furthermore, the generality of the questions may result in lower accuracy of information extraction.

To address these issues, we developed a technique for generating *dynamic questions*. Dynamic questions are generated automatically to find the relevant information of victims and suspects by using the names of subjects instead of general references such as "the victim" or "the suspect". Accordingly, the model searches for the relevant information only when victim or suspect names are found. This helps to create the dynamic question. For instance, if the model identifies the name of the victim as 'John Doe', then the dynamic questions composed by the algorithm are 'What was the age of John Doe?', 'What was the gender of John Doe?', 'What was the address of John Doe?'. Similar approaches are used to generate dynamic questions for finding the relevant information of the suspects as well. The main advantage of using dynamic questions is the reduced compute time and improved accuracy of information extraction.

Challenges The aforementioned process of information extraction gives rise to two major challenges. The first challenge is the longer length of the articles. The diversity of crawled news articles also consists of news articles with a longer length. The BERT QA language model can only process 512 tokens, and hence the articles with a higher number of tokens could not be processed. The second challenge is that the articles consist of multiple victims' names and multiple suspects' names. The model needs to extract information for each victim or suspect and their' respective information.

Challenge I: Processing Longer Articles

The longer articles that contain more than 512 tokens were processed by the model similar to the pooling process in CNN models. The article was broken into different pieces and then processed separately. The algorithm makes sure that the sentence in the article does not lose its semantic while breaking the article down into many pieces. For each information extraction for example finding the name of the victim, the model asks the respective set of questions to all the broken pieces. The first piece of the article is feed into the model with the set of questions. The algorithm will select the best answer with the best score for the first piece. The same process is repeated for other pieces as well. Each broken piece of article yields answers and a score. The algorithm then combines the answers and the scores yielded from all smaller pieces and then selects the answer with the maximum score. For each information extraction process, the respective sets of questions are asked to the model with the broken pieces of articles as the reference text, and later the answer with the best answer is selected.

Challenge II: Multiple Victims/Suspects Retrieval Method

The language model worked great to find the single name of the victim or suspect. However, there were always cases with multiple victims and multiple suspects. It was essential to extract the names of those all victims and suspects. The observation made in these cases was that the language model mostly gave the answer of a single victim/suspect even if multiple names were given. The reason might be that the semantic of the article where the victim was mentioned closely only referenced to the first name presented in the list of multiple names as the victim. The solution to finding the multiple victim/suspect name was the recursive algorithm. The algorithm finds the first name from the multiple names in the article, then removes that name from the articles, and then feeds that article to the model. The model yields a second name, the algorithm again removes that second name from the article and feeds that article back to the model to find the third name. The process continues until the name was found by the model. This approach successfully resulted in finding multiple victim/suspect names in the article.

The additional challenge arises when multiple victims/suspects name was to be found in a longer article. This problem was solved by combining both approaches. First, the articles were broken into smaller pieces. For each piece of the article, the algorithm finds the multiple victim/suspect using a recursive algorithm. And then combine the result at last by choosing the answer with the best score.

Experiment Setup

News Crawler Setup The news crawler was set up in the Amazon T2 Micro EC2 instance and was used to crawl the news for the duration of 4 days period. The search period can be changed by the users. The similarity checker algorithm parameters can be tuned by the users. The current threshold score is set at 50. For the cosine score more than 0.40 the total score of 40 is added and the cosine score more than 0.50, 50 is added to the total score, by default only articles with more than 0.30 scores are used for comparison. For person and location names, if there is a single names match between two articles, a score of 25 is added, if two names match a score of 35 is added and if three names match a score of 50 is added. For organization and quantity entities, if there is a single match score of 20, if two matches then a score of 30, and if three or more than that match's score of 50 is added. For example, two articles in which the title cosine similarity is 0.30 and has one name in common will have a total score of 55, which is greater than the threshold and hence two articles are similar. The news relevance checker consists of 20 keywords which are kidnapped, victim, survivor, abducted, rescued, trafficked, missing, sold, cheated, suspect, arrested, convicted, kidnapper, abductor, trafficker, agent, accused, caught, detained, and a broker. The final resulted news articles are saved in the MongoDB database.

Information Extraction Model Setup The information extraction model is composed of a text filter and the BERT QA language model. The BERT QA model being used is a pre-trained BERT Large Uncased model fine-tuned on SQuAD Dataset. The model class is set up to only give the answer which score is more than 0.90 otherwise to return 'Not Available'. The model was set up on Amazon ML T2 Large with 8GB Memory and 2 vCPUs. After the extraction of the information the MongoDB database is updated and also the CSV file on the Amazon S2 bucket is updated.

Evaluation Methodology The evaluation for both the News Crawler and the Information Extractor model was done manually. The news crawler output for the relevance of news articles was measured manually. For each group of relevant and non-relevant articles flagged by the crawler, the annotator went through each of the news manually to verify the results. The output from the Information Extraction model is not only a single word but sometimes consists of phrases, hence doing an automatic evaluation with string comparison was not feasible. In this case, also the annotator went through each article identifying victims and suspects and respective meta-data with additional information and then compared with the results generated by the model.

Results and Discussions

News Crawler Results The total number of news crawled in the four days news period was 1479. The crawler then processed these articles and then dropped the duplicate and similar articles. The remaining 463 articles were then passed to the news relevance checker where the algorithm flagged 122 articles as relevant and the 341 articles as non-relevant. After the manual evaluation, the total number of relevant articles was 184 and non-relevant articles were 279. The algorithm gave 45 false positives and 107 false negatives. The news relevance checker gave a precision score of 0.6311 and the accuracy score of 0.6717.

One of the important things that we need to focus on here is the false negative number because that 107 articles might contain the information on the victim/suspect. Our evaluation found that out of 341 non-relevant flagged articles there were 27 articles that contain the information on victim/suspect which was relevant to the objective. There are 3 reasons based on our observation on why those were flagged as irrelevant. The first reason is the lack of relevant keywords in the articles. The articles did not contain any of the 20 relevant keywords that will flag them as relevant. However, we can find the keyword in those missed articles and then add to the relevant keywords set but that will also start to flag other articles as relevant. The second reason is semantic of the articles which do not mention the suspect or victim in the context but later in the article provides the name of the victim/suspect. These articles with poor semantic and written styles were not only missed by the news relevance checkers but also performed poorly when feeding to the language model. The final reason is the human trafficking news being the high profile case and has its victim/suspect Wikipedia page. For example, the news article contains Jeffery Epstein as the suspect but since it contains the Wikipedia page, the algorithm flagged it as non-relevant.

The impact of the false-negative article should also be noted. If the articles contain any information about the victim and any search keywords related to human trafficking, then the model might yield suspect/victim giving wrong information. The total number of articles that did contain the information on suspect/victim and flagged as relevant was 77 out of 122 and 20 out of 122 articles containing the name of victims/suspects. The objective of the crawler is to find the human trafficking articles that contain the victim/suspect names and meta-data, but based on the observation most of the time the victims/suspects' names are kept secret in news articles. Hence the news article's narrative contains the relevant keywords but lacks the name of victims/suspects. We also need to note that this news was crawled in the duration of 4 days period during which the number of cases reported and covered by the media was limited, and also the cases in which victims/suspects' names appear might also be limited.

Information Extractor Results For the evaluation of the information extractor, we used a dataset of an annotated dataset of 366 relevant news articles. The average computation time for information extraction was 7.5 seconds per article. For the victim information, the model was provided with 142 articles that do contain victim names. The model matched the victim names in 103 articles and made mistakes in 39 articles giving an accuracy score of 72.53% for victim names prediction. The total number of victims that were presented in 142 articles was 347 out of which model extracted 205 victim names. For the evaluation of meta-data, we selected age and gender only since other information is not often presented in the articles. The victims' gender was matched in 93 articles and mistaken in 48 articles giving the accuracy of 65% in gender predictions. The overall gender mentioned was 222 out of which 197 were predicted correctly by model and 25 were mistaken. For the age of victims, the predictions of age were matched in 84 articles and mismatched in 58 articles giving 59.15% accuracy in age prediction. A total of 99 ages were mentioned out of which model predicted 66 of them correctly and made mistakes in 33.

In the case of identifying suspect names and meta-data the 230 articles containing suspect names were fed into the model. The model predicted suspects' names in 135 articles and made a mistake in 95 articles giving the accuracy of 58.69% in identifying suspect names. The total number of suspects presented in 230 articles was 600 out of which the model extracted 222 names. For the suspect gender, the model predicted the gender in 114 articles correctly and made mistakes in 116 articles giving the accuracy of 49.35%. The total number of gender given were 235 out of which the model predicted 165 of them and made the mistake of identifying 70 genders. For suspects' gender, the model predicted the gender correctly in 115 articles and made mistakes in the other 115 articles giving an accuracy of 50%. The total number of ages given was 91, and the model predicted 60 out of 91 correctly and made mistakes in predicting the other 30 pages.

We need to evaluate and discuss cases of victims and suspects separately since both of them are vastly different in terms of their significance, occurrence in articles, and identifying process. The question that is used to identify victim and suspect names are different, the number of victim names given in the article is less than the number of times suspect names are given in the article. The accuracy is measured in terms of the number of matching results in the articles. And also the number of successful extraction for names and meta-data is provided to make an analysis of how the model works in identifying and extracting information from the articles.

The model performed well in the case of identifying the victims in articles containing a single name. In the articles with multiple names in them, the overall model makes use of a recursive algorithm to search for names, when the first name is given by the model then the model removes that name from the articles and sends it back to the model to find the next name and repeat the process. This process distorts the semantic of the articles after removing names. If the names of victim/suspect are given in the list with age on its side, removing the name will leave the age only in the sentence that will distort the semantic of the whole sentence including the article. For example, if the multiple victim's names are represented in this pattern in articles 'the victims are John Doe 28, Mary Lith 30, William James 29', removing the second name after the second recursion will leave the sentence to be 'the victims are 28, 30, William James 29'. In cases like this model miss identifying a few names that appear at last in the list. The model also made mistakes in identifying the names of victims and suspects. There have been cases where models predicted the victim's names instead of suspects and vice versa. The reason for this is because of the written style of the articles. The articles do not mention the victim or the suspect's name with explicit use of relevant keywords. In some cases, the articles talk about the victim being abducted in the first few sentences and present the name of the victim at almost the end of the articles. Hence the model misses the name or mixes the other name. However, this problem has been addressed in the model, for each extraction of the victim name model cross-check with a suspect list, if the name is presented in the suspect list, the name of the victim is then shifted to the flagged victim list. The same process is adopted for the extraction of the suspect names as well.

The identification of gender for victims and suspects depends upon the context in which names are mentioned in the articles. In the case of multiple victims/suspects, gender is missed in large cases. Hence, the model makes use of Gender-API [8]. If the gender is not recognized for the name by the model then it sends the name to Gender-API to detect the gender. The challenge here is that few of the names are native to that particular country and are very unique in such cases that even the Gender-API cannot determine the gender. The age in articles is presented in small numbers. The age is also represented in an ambiguous way such as victim in twenties or suspect in forties. The model cannot extract these ambiguous terms. The ages are sometimes mentioned in the word and the algorithm converts them into digits. In a few cases model mistakes other numbers for the age of the suspect/victims.

Conclusion

Our work demonstrates the use of recent advances in natural language processing and machine learning to create an automated pipeline to streamline the curation, analysis, and extraction of actionable intelligence from multi-sourced news media as Open-Sources of Intelligence (OSINT). We also demonstrated that the use of BERT Question Answering model can be used for important information extraction from unstructured data and can leverage the model with additional algorithms to reduce the computation cost and redundant processing by only curating relevant news articles. The performance of the overall model and the news crawler demonstrate that the developed system can be used for information extraction for human trafficking cases. This is the first phase of a more comprehensive data collection pipeline and framework for OSINT on human trafficking. The future work includes enhancing our system to detect patterns and techniques to find the presence of human trafficking in real-time. Additionally, future work also includes addressing the limitation of the system to search and process articles in the English language by using automated translation tools.

The proof-of-concept code are made available on GitHub[9].

Acknowledgments

This work was supported by Love Justice International. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Love Justice International. The authors are only responsible for developing the system and tools for data extraction but not for the usage of data.

References

Brewster, B.; Ingle, T.; and Rankin, G. 2014. Crawling open-source data for indicators of human trafficking. In *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, 714–719.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

Gibson, S. 2004. Open source intelligence. *The RUSI Journal* 149(1):16–22.

Grootendorst, M. 2020. Keybert: Minimal keyword extraction with bert.

Hundman, K.; Gowda, T.; Kejriwal, M.; and Boecking, B. 2017. Always lurking: Understanding and mitigating bias in online human trafficking detection. *CoRR* abs/1712.00846.

[8] https://gender-api.com/

[9] https://github.com/UNHSAILLab/Combating-Human-Trafficking-via-Automatic-OSINT-Collection

Kejriwal, M.; Ding, J.; Shao, R.; Kumar, A.; and Szekely, P. A. 2017. Flagit: A system for minimally supervised human trafficking indicator mining. *CoRR* abs/1712.03086.

Mensikova, A., and Mattmann, C. 2017. Ensemble sentiment analysis to identify human trafficking in web data.

Quick, D., and Choo, K.-K. R. 2018. Digital forensic intelligence: Data subsets and open source intelligence (dfint+osint): A timely and cohesive mix. *Future Generation Computer Systems* 78:558–567.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR* abs/1606.05250.

Shi, P., and Lin, J. 2019. Simple BERT models for relation extraction and semantic role labeling. *CoRR* abs/1904.05255.

Tong, E.; Zadeh, A.; Jones, C.; and Morency, L.-P. 2017. Combating human trafficking with multimodal deep models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1547–1556. Vancouver, Canada: Association for Computational Linguistics.

Yang, W.; Xie, Y.; Lin, A.; Li, X.; Tan, L.; Xiong, K.; Li, M.; and Lin, J. 2019. End-to-end open-domain question answering with bertserini. *CoRR* abs/1902.01718.

Yang, W.; Zhang, H.; and Lin, J. 2019. Simple applications of BERT for ad hoc document retrieval. *CoRR* abs/1903.10972.

Zhu, J.; Li, L.; and Jones, C. 2019. Identification and detection of human trafficking using language models. In *2019 European Intelligence and Security Informatics Conference (EISIC)*, 24–31.