Crime Science

RESEARCH

Open Access

Supporting crime script analyses of scams with natural language processing

Zeya Lwin Tun[1*] and Daniel Birks[2]

Abstract

In recent years, internet connectivity and the ubiquitous use of digital devices have afforded a landscape of expanding opportunity for the proliferation of scams involving attempts to deceive individuals into giving away money or personal information. The impacts of these schemes on victims have shown to encompass social, psychological, emotional and economic harms. Consequently, there is a strong rationale to enhance our understanding of scams in order to devise ways in which they can be disrupted. One way to do so is through crime scripting, an analytical approach which seeks to characterise processes underpinning crime events. In this paper, we explore how Natural Language Processing (NLP) methods might be applied to support crime script analyses, in particular to extract insights into crime event sequences from large quantities of unstructured textual data in a scalable and efficient manner. To illustrate this, we apply NLP methods to a public dataset of victims' stories of scams perpetrated in Singapore. We first explore approaches to automatically isolate scams with similar modus operandi using two distinct similarity measures. Subsequently, we use Term Frequency-Inverse Document Frequency (TF-IDF) to extract key terms in scam stories, which are then used to identify a temporal ordering of actions in ways that seek to characterise how a particular scam operates. Finally, by means of a case study, we demonstrate how the proposed methods are capable of leveraging the collective wisdom of multiple similar reports to identify a consensus in terms of likely crime event sequences, illustrating how NLP may in the future enable crime preventers to better harness unstructured free text data to better understand crime problems.

Keywords Scams, Crime, Policing, Crime script analysis, Unstructured data, Natural language processing, Term frequency-inverse document frequency, Doc2Vec

Introduction

Scams have become increasingly prevalent alongside greater Internet connectivity and ubiquitous use of digital devices. Any person around the world can be a potential victim. Scams are as much a cause of concern in Singapore as they are globally. Efforts by authorities to combat scams include the establishment of the Anti-Scam Centre in 2019, regular police operations against domestic and transnational syndicates as well as public education campaigns. Despite such efforts, victims continued to fall prey to scams, evident from the increasing number of reported scam cases over the last few years, from 9,502 cases in 2019, to 15,756 cases in 2020 and to 46,196 cases in 2021 (Singapore Police Force, 2020a, 2020b; Lin, 2022).

Scams impact victims in numerous ways. The most immediate consequence is financial loss, which in turn causes tremendous emotional stress, particularly if significant amounts of savings were involved. Victims also experience embarrassment, shame and humiliation, especially in the case of love scams (Buchanan & Whitty, 2014). Scams also have longer-term psychological effects on victims, such as increased anxiety and low self-esteem. Additionally, scams that involve loss of personal

*Correspondence:
Zeya Lwin Tun
zeya.zlt@gmail.com
[1] School of Mathematics, University of Leeds, Leeds, UK
[2] School of Law, University of Leeds, Leeds, UK

data and misuse of identity can prolong the distress faced by victims by enabling subsequent victimisation. Given the gravity of social, psychological, emotional and economic impacts that scams bring about, there is a strong rationale to enhance our understanding of scams in order to devise ways in which they can be disrupted.

One potential method for increasing understanding of scams, and crime events more generally, is script analysis. Script analysis was first conceptualised for application on crimes in 1994, with the objective of studying the sequence of events prior to, during and after the commission of a crime (Cornish, 1994). Crime scripts provide an organising framework to understand the modus operandi involved in crime events, in this case scams, and thereby identify points of intervention where they might be disrupted and ultimately prevented. Conducting crime script analyses, however, requires a considerable amount of effort. The first step generally involves collecting data from a variety of sources, including open-source and confidential documents, before manually analysing the data to extract key information about the different stages of the crime event. These processes are exceedingly resource-intensive. Given the significant effort involved, there is huge potential to leverage Natural Language Processing (NLP) methods to alleviate the effort required, particularly if the source data is in textual form.

In this paper, we present research exploring how NLP methods can be used to support crime script analysis. In particular, we seek to apply NLP methods on free text stories of scams extracted from 'Scam Alert', a Singapore-based website aimed at promoting scam awareness, to generate 'scam scripts'. Here, we define a scam to be a scheme designed to deceive individuals into giving away their money or personal information, generally using the Internet or some other communication medium such as a phone call or text message. Our study entails two distinct but interrelated stages. First, we develop methods for identifying scam reports with similar modus operandi. Second, we implement approaches capable of generating key words or phrases, as well as sequences of actions from these similar reports that reflect characteristics of a given modus operandi. These characteristics can then be thought of as different steps in the scam script. Given that our text corpus from 'Scam Alert' contains accounts on a wide variety of scams, and script analyses inherently focus on specificity in understanding the commission of crime, the first stage was a necessary step so as to isolate similar scam reports, thus maximising the likelihood that subsequent analyses would produce high quality scam scripts. Collectively, the methods outlined in this paper provide new and efficient means for supporting scam script analyses.

The remainder of this paper is organised as follows. First, we briefly review academic literature related to crime script analyses as well as the applications of NLP on free text in the domains of crimes and scams. Subsequently, we provide a theoretical overview of the various NLP techniques that were used in this research. The next sections outline steps taken to extract, clean and pre-process the text data from 'Scam Alert' and describe our methodology in identifying scam reports with similar modus operandi as well as generating key words and phrases from them. We then illustrate this approach through a case study of analysing High Court impersonation scams. Finally, we summarise our key findings, highlight its limitations, and propose areas of future research.

Related work

Why scammers succeed?

To understand how scams fundamentally work, it is useful to examine past studies relating to scams from various theoretical perspectives. One such perspective is the Routine Activity Theory (Cohen & Felson, 1979), which states that crime requires the convergence of a motivated offender, suitable target, and the absence of a capable guardian. While originally conceived to describe direct contact predatory offending, one might well argue that this theory is applicable in the context of scams. Much research has been conducted to study the attributes of victims that might make them susceptible to scams. Amongst these are low self-control (Wilsem, 2011), impulsiveness (Pattinson et al., 2011), perception towards the size of reward (Fischer et al., 2013) as well as loneliness (Buchanan & Whitty, 2014). Perhaps surprisingly, there have been mixed findings about how an individual's familiarity with technology and the Internet influences their susceptibility to scams. While Wright and Marett (2010) found that being more Internet-savvy led to lower susceptibility, Wilsem (2011) observed that Internet use per se did not protect victims regardless of their savviness. For instance, in a study by Vishwanath (2015), e-mail habits were found to increase chances of phishing victimisation—one might well view this finding through the lens of Routine Activity Theory, such that those who check their e-mails regularly are potentially suitable targets as they tend to be more likely to click on hyperlinks in phishing e-mails.

Scams typically require a degree of planning to be successfully executed. For this reason, the Rational Choice Perspective (Cornish & Clarke, 1986) might also be applied to those who perpetrate scams, or in popular parlance—"scammers". It is clear that scammers make rational choices when committing scams, including actions taken to increase chances of success. For instance, scammers invoke visceral appeals from victims, including

Table 1 Script analysis of an online purchase scam

Stage	Action
Preparation	Victim collects information on iPhone 13, such as its price and features
	Victim becomes interested to purchase iPhone 13
	Offender puts up a fake listing of iPhone 13 online
Pre-activity	Victim comes across the listing of iPhone 13 put up by offender
	Victim believes the listing of iPhone 13 and the seller are genuine
	Victim compares prices across other similar listings
	Victim contacts offender to discuss modes of payment and delivery
Activity	Offender tells victim that the listed price is for limited time only
	Victim decides to purchase iPhone 13 from offender
	Offender requests victim to make a bank transfer
	Offender promises victim delivery of item within 3 days
Post-activity	Victim fails to receive item within 3 days
	Victim contacts offender to check on the delivery
	Offender does not respond to victim

appeals to love, in the case of love scams (Yee et al., 2019), and authority, in the case of scams impersonating government officials (Friedman, 2020). They expend effort to design messages that mirror official communications or require an urgent response in order to increase rates of response from victims (Yee et al., 2019; Wang et al., 2012; Luo et al., 2013).

A lack of capable guardianship is another factor contributing to the success of scams. In their study, Graham and Triplett (2017) used digital literacy as a measure of guardianship and discovered that respondents with higher digital literacy reported receiving phishing emails more. Their study suggested that higher levels of guardianship meant greater awareness of phishing. Another study found that remote and regional communities in Australia which were disadvantaged in terms of receiving "adequate levels of capable guardianship" were associated with higher fraud risks (Smith & Jorna, 2011).

Crime script analysis

Key to the prevention of crime is understanding the "how" of crime events – in this case that is, how scams happen. One way to do so is through crime script analysis, which Cornish (1994) first introduced as "a way of generating, organising and systemizing knowledge about the procedural aspects… of crime commission." More recently, crime scripts were defined by Ekblom and Gill (2016) as "abstracted descriptions of a particular kind of behavioural process, namely structured sequences of behaviour extended over time and perhaps space." There have been limited guidance on how crime scripts should be created, as pointed out by Brayley, Cockbain and Laycock (2011), although there have been some attempts to develop structured processes, such as the recent work by Chainey and Berbotto (2021). Notwithstanding, Dehghanniri and Borrion (2021), in their systematic review of crime scripting, questioned the "usefulness of formalizing the crime scripting process". Having said that, there is some general consensus on the characteristics that crime scripts should possess. Fundamentally, they must describe the sequence of activities involving all relevant parties before, during and after a crime (Cornish, 1994). Tompson and Chainey (2011) proposed a simple model to categorise those activities into four stages—preparation, pre-activity, activity and post-activity. In addition, crime scripts may be represented as a flowchart or in a tabular format, where each row corresponds to a specific activity (Dehghanniri & Borrion, 2021). Where feasible, each activity should also be described using a consistent syntax, for instance, subject–verb–object (Borrion, Dehghanniri, & Li, 2017). Table 1 illustrates how a script could look like based on a hypothetical example of an online purchase scam.

Since the 1990s, crime script analysis has gained significant traction within the research community, evidenced by an increase in the number of publications related to crime scripts over time (Dehghanniri & Borrion, 2021). Over the years, this approach has been utilised in developing prevention strategies for a range of criminal offences. Examples include child sex trafficking (Brayley, Cockbain, & Laycock, 2011), terrorism (de Bie et al., 2015; Osborne & Capellan, 2016) and methamphetamine manufacturing (Chiu et al., 2011). The application of script analysis on scams is also not new. Choi, Lee and Chun (2017) used script analysis to dissect voice phishing scams in South Korea and in a more recent

work by Nguyen (2021), script analysis was used to gain a better understanding of two types of transnational computer fraud in Vietnam, namely bank card data fraud and phone scams. While our work does not deal with developing scripts for any specific type of scam per se, it aims to facilitate the development of scam scripts using NLP, with the eventual goal of identifying points of intervention and measures where scams can be disrupted.

It is also important to note that the vast majority of previous studies have performed script analyses from offenders' perspectives (Leclerc, 2014)—that is, identifying the actions an offender must undertake to successfully commit a crime. This focus is the result of a desire to identify means to disrupt offender behaviour. This also reflects the sources of data commonly used in script analyses, such as police investigation files, media account and legal documents, which often focus on offenders. However, as the Routine Activity Theory depicts, there are other actors beyond offenders involved in a crime commission process, namely victims and guardians. Leclerc (2014) asserts that it is equally important to examine scripts from the points of view of the victims and guardians alongside. Leclerc (2014) also introduced the idea of 'interpersonal scripts', which capture interactions between the offender's scripts, the victim's scripts and the guardian's scripts.

In this paper, we take the approach of exploring victim's scripts of scams, doing so for several reasons. First, pragmatically the text data describing scams analysed in this paper contains victims' stories of scams. Second and relatedly, beyond the scope of this study, gaining access to sufficient quantities of reliable data with regard to the perpetration of scams is challenging, whereas victims' accounts of scams—such as those analysed here—are relatively commonplace and publicly available—thus presenting a unique opportunity to explore how NLP may support the crime-scripting process. Third and most importantly, given the widespread nature of online scams, identifying the behaviour of victims that lead to successful or unsuccessful scams has the potential to inform crime prevention efforts that could be targeted at those most likely to become victims.

NLP applications for crimes and scams

The application of NLP and machine learning methods on unstructured free text in the domains of crimes and scams is not novel, with a range of methods having previously been explored. To illustrate, Named Entity Recognition (NER), an NLP task that picks out entities such as names and locations from a given text, has previously been used to extract entities to construct meaningful crime networks that could support investigations and identify criminal links (Al-Zaidy et al., 2012; Elyezjy & Elhaless, 2015). Similarly, through collaborations between the Dutch National Police and Utrecht University, information extracted from crime reports using NER were used in a formal reasoning system. This system enabled automatic formulation of questions, which could be posed to complainants to provide clarification on their reports (Schraagen, Testerink, Odekerken, & Bex, 2018).

Classification is another common NLP task in the crime domain. In their study, Mbaziira and Jones (2016) used machine learning algorithms such as Support Vector Machines, Naïve Bayes and k-Nearest Neighbours on free text in e-mails to classify and detect phishing scams. Mbaziira and Jones (2016) also experimented with textual data collected from Facebook profiles linked to known cyber-criminals to predict fraudulent profiles. In a separate work, Mbaziira used bilingual text datasets in English and Nigeria Pidgin to detect phishing scams (Mbaziira et al., 2015). More recently, Hamisu and Mansour (2020) developed a classifier to detect e-mails relating to possible advance-fee scam using a bag-of-words model. More state-of-the-art NLP approaches have also been used within the crime domain. For example, Naudé, Adebayo and Nanda (2022) experimented with transformer models such as Bidirectional Encoder Representations from Transformers (BERT) to detect fraudulent job advertisements.

Another NLP approach previously applied in the study of crime is topic modelling. Topic models use statistical machine learning to identify latent topics within collections of free text documents. In their work, Kuang et al. (2017) used non-negative matrix factorisation to discover topics within free text crime reports associated with multiple offence types, their goal being to explore meaningful latent crime classes that can be derived from examining descriptions of crimes. In a related work, Birks et al. (2020) used another topic modelling algorithm known as Latent Dirichlet Allocation (LDA) that assumes each document to contain a mix of different latent topics. Utilising this technique to examine topics within a single crime classification—residential burglary, Birks et al. (2020) explored if LDA could aid in the identification of similar burglary modus operandi which are likely to require specific crime prevention interventions. Both studies resonate strongly with research presented in this paper because of the common motivation to leverage NLP methods to analyse free text descriptions of crime events in order to better understand the processes of crime commission.

Thus far, we have seen how NLP methods had been experimented with free text such as crime reports, investigation documents and e-mails. A less ubiquitous type of textual data on which NLP methods had also been applied in literature is Hypertext Markup Language

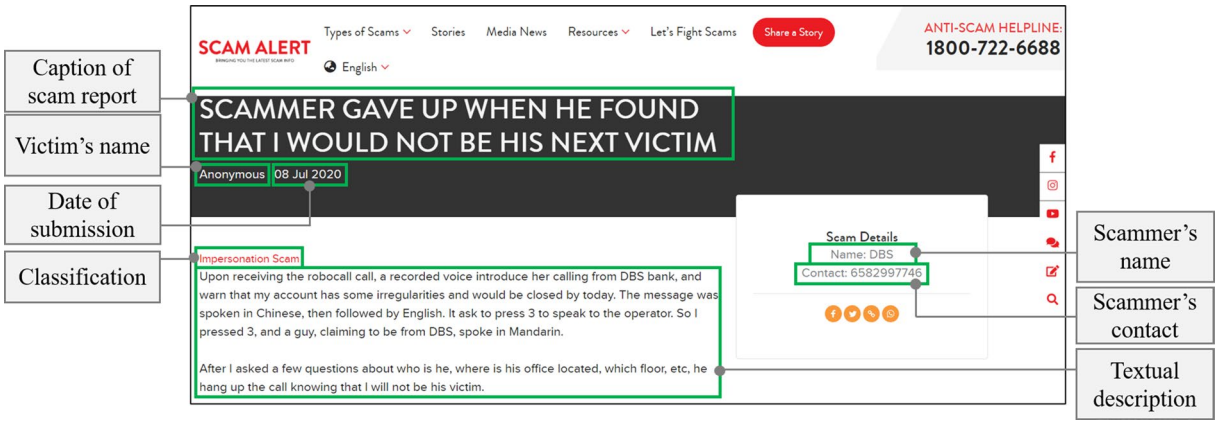

Fig. 1 A screenshot of a scam report published on 'Scam Alert'

(HTML), the standard language for displaying web pages. Given the prevalence of scams perpetrated via fake or fraudulent websites, there is understandably an interest to detect them using analytical approaches. In their work, Drew and Moore (2014) sought to automatically identify and link replicated scam websites together by employing hierarchical clustering algorithms as well as Jaccard similarity on various attributes of websites, including their HTML tags. In a similar vein, Phillips and Wilder (2020) examined the typology of advance-fee cryptocurrency scams by applying another clustering technique known as density-based spatial clustering of applications with noise (DBSCAN) on HTML data.

Evidently, NLP and machine learning methods have been extensively applied in the domains of crimes and scams for a wide range of use-cases. However, to the best of our knowledge, no previous studies have explored the use of NLP methods in creating scam scripts, or more generally crime scripts. The study that we found somewhat similar to ours in terms of encompassing a comparative analysis of multiple textual descriptions was that by Borrion, Dehghanniri and Li (2017). Even then, their work involved manually comparing textual *scripts* of the *same* crime event, whereas ours uses NLP methods to analyse textual *reports* of *different* scams. We also observe from current literature that the data sources used to generate crime scripts had been predominantly textual in nature. Examples include text messages (for example, Brayley, Cockbain, & Laycock, 2011), case summaries (for example, de Bie et al., 2015), transcripts of interviews with investigators (for example, Choi, Lee, & Chun, 2017; Nguyen, 2021), court transcripts (for example, Chiu et al., 2011) and investigation documents (for example, Osborne & Capellan, 2016). The methods used to generate crime scripts were also varied. For instance, Nguyen (2021) used an inductive approach to analyse modus operandi of transnational computer fraud while Chiu et al. (2011) as well as Osborne and Capellan (2016) coded their textual data according to various domain-specific categories. Regardless, the methods used were highly manual, and understandably resource-intensive. Therefore, the work presented in this paper aims to reap the benefits of NLP by not only alleviating the effort required in developing scripts but also achieving consistency and scalability from large quantities of textual data.

Data preparation and analytical approaches

We now proceed by describing our primary data source, the procedures taken to extract, clean and pre-process it, and the several NLP techniques applied in its analyses.

Data extraction and cleaning

The data used in our research was derived from 'Scam Alert', a website launched by the National Crime Prevention Council (NCPC) of Singapore in 2014. 'Scam Alert' contains resources about different types of scams as well as tips to avoid becoming a victim. The website also offers members of the public a place to share stories of their encounters with scams with the hope of preventing others from being victimised. Figure 1 shows a screenshot of a published scam report containing various information including the textual description of the scam provided by a user of the website. It is these free text data that are analysed in this study.

To use scam reports from 'Scam Alert' for our research, we first obtained written permission from the data owner, NCPC (Poh, personal communication, 2020). Web-scraping techniques were then applied using the Python libraries, 'BeautifulSoup' (Richardson, 2007) and 'Selenium' (Software Freedom Conservancy, 2013), to extract information such as date of submission, textual description, scammers' details and scam category. A total

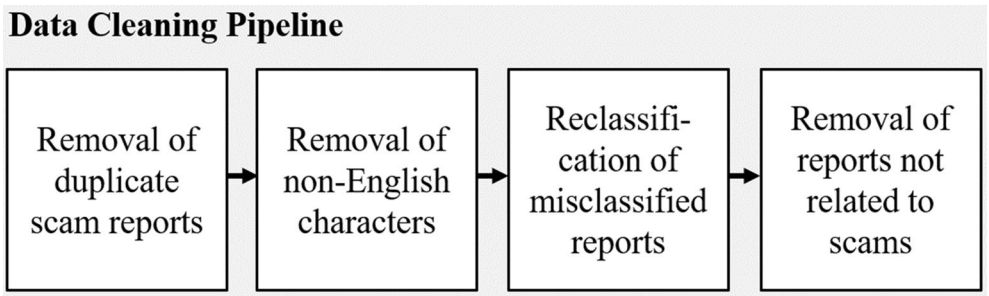

Fig. 2 Data cleaning pipeline

of 4,660 scam reports were extracted, describing scams that took place over a period of four years (from 20 July 2016 to 19 July 2020).

Once extracted, these data were cleaned. Our data cleaning pipeline is summarised in Fig. 2. We began by removing 64 duplicate scam reports. Following this, we checked for reports containing non-English characters. Out of 96 scam reports that contained non-English characters, six were entirely in Chinese characters and were removed. For the remaining 90 reports, only the Chinese characters were removed since they occupied only a small segment of each text.

The next steps in the data cleaning process were reclassifying misclassified scam reports and removing reports unrelated to scams. These were done by inspecting scam reports and assessing whether the original scam category matches the described stories, according to the definitions of various scam types on the 'Scam Alert' website. Given the laborious task involved to manually inspect each report, we inspected a sample of 1232 scam reports, of which 256 were found to be misclassified and 36 unrelated to scams. The misclassified scam reports were reclassified into the most appropriate categories, whereas those unrelated to scams were removed. The remaining 4554 scam reports formed the text corpus for our research. These 4554 scam reports spanned across 21 distinct scam types, the most common ones being impersonation scam, online purchase scam and internet love scam. The average length of scam reports was about 99 words, with a standard deviation of 85 words, suggesting a huge variance in the lengths of the reports.

Text pre-processing

Free text, in its raw form, is highly unstructured and noisy. Text pre-processing is therefore an essential step in any NLP task to make text more consistent and interpretable to machines. This current section elaborates on our text pre-processing pipeline, as represented in Fig. 3.

The first step involved removing unicode characters from the text. Though not prevalent, the text corpus contained several unicode characters embedded within words. For instance, "n\u200cext" was the characters "\u200c" embedded within the word "next". Given that such words would be of utility in our corpus, we removed only the unicode characters.

Next, digits such as dates, times and telephone numbers are typically regarded as noise in any textual data and removing them helps towards standardising the text. Moreover, for the purposes of crime scripting in this research, we are interested in capturing the meanings of scam reports from their text and digits were assessed to play an insignificant role in this regard. Across all scam reports, a total of 16,468 strings were found to contain digits and were thus removed. The next pre-processing step involved expanding contracted words, such as "don't" and "can't". Other steps to help standardise text in our corpus included converting words into their lower cases as well as removing white spaces and punctuation marks. The next step was to unabbreviate acronyms. The general approach to find acronyms in our text was using regular expressions to detect consecutive upper-case letters. An example was "IBAN", which represents the phrase "International Bank Account Number". Using this approach. a total of 3,495 acronyms across 63 unique acronyms were found and replaced with their unabbreviated forms. Acronyms which were written in lower-case letters were identified manually.

Given that our text corpus originated from victims with different language proficiencies, it contained a range of typographical errors. Rectifying these errors was another crucial step in our text pre-processing.

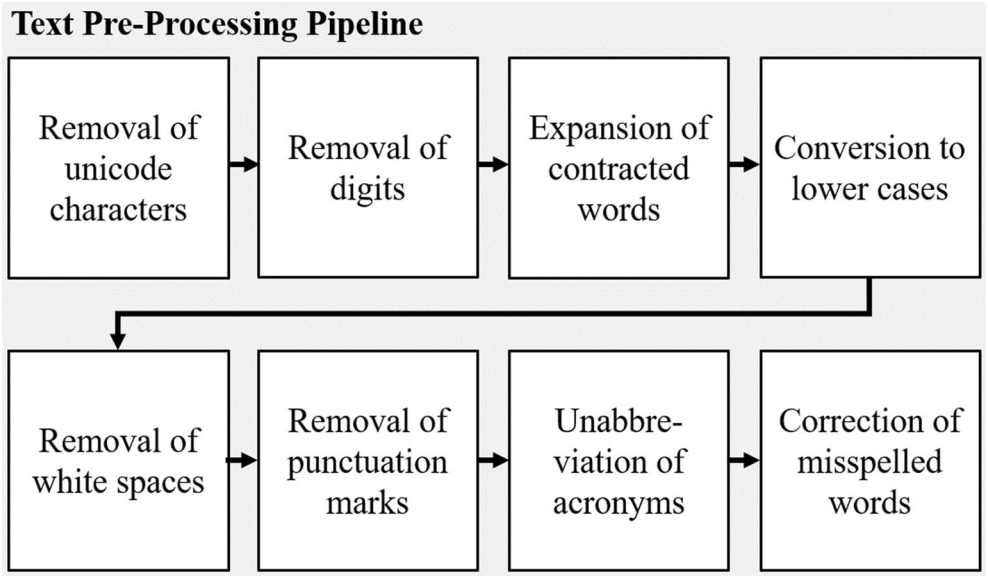

Fig. 3 Text pre-processing pipeline

Unlike an acronym which corresponded to a unique phrase, words were misspelled in several different ways. A dual approach was adopted to rectify such errors. The first involved using 'pyspellchecker', a Python library that identifies misspelled words using the Levenshtein[1] distance algorithm (Norvig, 2018). However, 'pyspellchecker' was unable to detect typographical errors of lesser-known words or words unique to our text corpus. The second approach thus involved manually and iteratively identifying misspelled words during the course of our research,[2] which enabled us to rectify the spellings of 1,099 words across the corpus.

Natural language processing techniques

In analysing these cleaned free text data, we explore the use of several distinct NLP techniques with the aim of supporting crime script analyses. These techniques are now briefly introduced and explained for readers who may be unfamiliar with them.

Doc2Vec

In 2013, Mikolov, Chen, Corrado and Dean (2013) developed the *word-to-vector* (Word2Vec) algorithms, which generate vectors of numbers that represent words and their semantics. The *document-to-vector* (Doc2Vec) algorithms, developed by Mikolov and Quoc in 2014, extends Word2Vec from the word level to the document level (Le & Mikolov, 2014). In essence, Doc2Vec is an NLP algorithm that learns numeric "fixed-length feature representations" of text in documents.

There are two types of Doc2Vec algorithms: *Distributed Memory model of Paragraph Vector* (PV-DM) and *Distributed Bag-of-Words version of Paragraph Vector* (PV-DBOW). Both algorithms were used in this research. To illustrate how they work, consider the following text which describes how an individual purchased a product they never received from the Singapore-based online marketplace, Carousell:

> *"I bought headphones on Carousell but never received them."*

In both frameworks, every document and every word in the document are each mapped to a unique vector. Figure 4 is a simple illustration of the PV-DM algorithm for the first five words of the above text (that is, a window size of five). Here, the target word is "headphones" and the context words are "i", "bought", "on" and "carousell". A vector corresponding to the document (herein referred to as "Tag ID #22"), together with vectors representing the context words, are trained to predict the target word. This is done using an artificial neural network (ANN). The window of text slides over the text and the process repeats – thus, where the target word is "Carousell", context words are "headphones", "on", "but" and "never".

The PV-DBOW algorithm is illustrated in Fig. 5. Instead of using both the vectors for the document and the vectors for the context words, PV-DBOW only uses the former in training. It trains the document vector to predict words within a window of text using an ANN. In this example, the document vector, Tag ID #22, is trained

[1] In the Levenshtein distance algorithm, permutations of words within an edit distance of two were compared.
against known words (Norvig, 2018).

[2] While this manual approach may not scale well to larger corpus, we deemed it appropriate in the context of this research to explore what could be derived given a high-quality dataset.

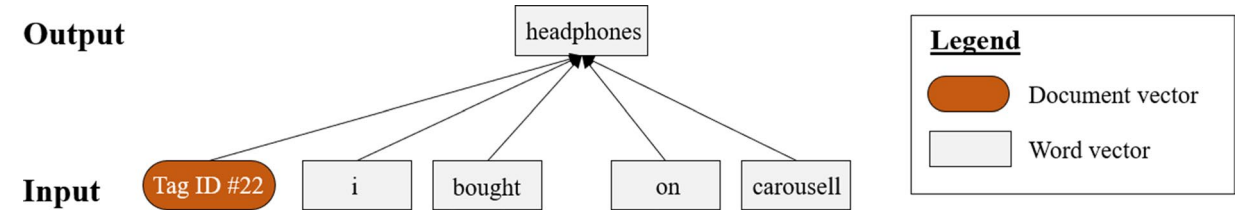

Fig. 4 Illustration of PV-DM algorithm of Doc2Vec

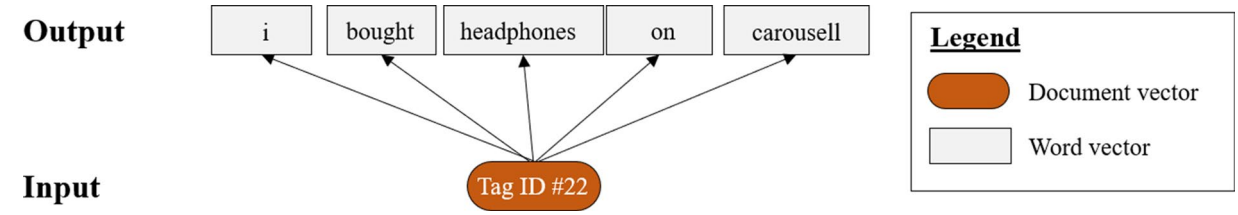

Fig. 5 Illustration of PV-DBOW algorithm of Doc2Vec

to predict the first five words of the text. Similarly, the training is repeated for different windows of words across the text.

At the end of training, both algorithms have learned numeric representations of the document, also known as a *document vector*. These trained Doc2Vec models can also perform inferences to compute document vectors for new and unseen documents.

Cosine similarity

By representing documents numerically as document vectors, useful semantic properties, such as document similarity, can be discovered. One of the metrics used in this research to measure similarity between document vectors corresponding to scam reports was *cosine similarity*. It is defined by

$$\cos\theta = \frac{\mathrm{X} \cdot \mathrm{Y}}{|\mathrm{X}||\mathrm{Y}|}$$

where X and Y are document vectors, and θ is the angle between them. If document vectors X and Y are perfectly similar, θ is 0° and the cosine similarity score is 1. Conversely, if they are perfectly dissimilar, θ is 180° and the cosine similarity score is -1. Put another way, the closer the cosine similarity score is to 1, the more similar two documents are.

Jaccard similarity

While cosine similarity operates in the vector space and measures similarity between two document vectors, Jaccard similarity deals with the 'text space' and measures similarity between two documents in terms of the proportion of words they have in common. Jaccard similarity was the alternative approach used in this research to find similar scam reports. Mathematically, it is given by the expression,

$$\mathrm{Jaccard}(\mathrm{A}, \mathrm{B}) = \frac{|\mathrm{A} \cap \mathrm{B}|}{|\mathrm{A} \cup \mathrm{B}|}$$

where A and B represents sets of all words in two documents. A Jaccard similarity score of 1 means that two documents have exact same words, whereas a Jaccard similarity score of 0 means they have no words in common. The higher the Jaccard similarity score, the more words two documents have in common.

Term frequency-inverse document frequency

In our research, the *Term Frequency-Inverse Document Frequency* (TF-IDF) method was used to generate key words and phrases (collectively referred to as "terms") from a set of similar scam reports. More generally, TF-IDF is a method that scores each term in a set of documents according to how unique it is. It does this by taking into consideration the importance of a term in a single document, and then scaling it by the importance of that term across all documents. Mathematically, the TF-IDF score for term *i* in document *j* is given by

$$\mathrm{w}_{\mathrm{i,j}} = \mathrm{TF}_{\mathrm{i,j}} \times \log\left(\frac{\mathrm{N}}{\mathrm{d}_{\mathrm{i}}}\right)$$

where $\mathrm{TF}_{i,\,j}$ is the frequency of term *i* in document *j* (also known as *Term Frequency*), N is the total number

of documents and d_i is the number of documents that contain term *i*. The latter term in the right-hand side of the above expression corresponds to the *Inverse Document Frequency* (IDF) of term *i*. The less frequent a term *i* appears across the documents, that is, the smaller the value of d_i, the higher the IDF and therefore the TF-IDF score for term *i*. In practice, TF-IDF provides a reliable and effective method in discriminating terms which are unique to a set of documents from those which are not.

Methodology

Our work exploring how NLP can support crime script analyses involved two interrelated stages: first, exploring how NLP methods can identify scam reports with similar modus operandi; and second, using NLP to extract key words, phrases and potentially sequences of actions from those similar reports with the aim of providing useful insights for the generation of crime scripts. In this section, we explain the motivation behind each stage and specify the methodology involved.

Finding similar scam reports

Before generating key words and phrases that will be helpful towards script analyses of a particular scam report, it is first necessary to find other similar scam reports. There are several ways of defining similarity of scam reports. Depending on use-cases, scam reports can be said to be similar if they make mention of the same unique identifiers such as bank account and telephone numbers, share common entities like monikers and names of online platforms or pertain to similar time periods or sequence of events. In our work, as far as crime script analysis is concerned, it was of interest to define similarity in terms of the modus operandi of a scam. Our hope here is that the key words and phrases that will be generated from a collection of similar scam reports will accurately capture the key ingredients and procedural elements of a given modus operandi rather than grouping reports of what are likely to be the same scam (something that might be seen as overfitting in this context). This desire to identify similar but not just the same scams we propose will best support those who seek to construct reliable crime scripts for the type of scam in question (this distinction is somewhat analogous to the difference between crime linkage—the identification of crimes perpetrated by the same offender(s)—and techniques designed to detect offences with similar modus operandi).

To capture similarity in modus operandi, it was also not sufficient to use the previously discussed discrete scam categories which users of the website use to tag each scam report. Scam categories are in essence discrete labels that describe high-level characteristics of scams (for example, Impersonation Scam). They do not provide procedural insights into how scams happen. In contrast, modus operandi of scams includes more defining characteristics, such as how scammers first made contact with victims, the deception used by the scammer or the mode by which victims lost their monies. Indeed, given the variety of scams reported on 'Scam Alert', each scam category encompasses many different modus operandi. Therefore, there is a need to move beyond selecting scam reports of the same scam category, to developing a methodology that isolates scam reports with a specific modus operandi. Moreover, the better our process in isolating scam reports with similar modus operandi, the lesser the amount of noise amongst the collection of similar scam reports, and the better the quality of any resulting analyses designed to support crime scripting. In the following sub-sections, we present two approaches in which similarity of scam reports in terms of modus operandi could be quantified: the vector-based and text-based approach.

Vector-based approach

Our first approach in measuring similarity of modus operandi in scam reports is the vector-based approach, which utilises Doc2Vec document vectors and cosine similarity. Doc2Vec was selected over other NLP methods for several reasons. First, the approach seemed well-suited to our narrative-like scam reports—taking sequence of words in a document into account (unlike the bag-of-words approach) and generalising to texts of different lengths. Moreover, Doc2Vec has been used in many previous studies, is relatively straightforward to implement, and does not require labelled data for training—with a trained model being able to produce vectorised representations of scam reports in a relatively simple manner.

In order to encode scam reports as document vectors, we used the pre-processed scam reports following data preparation steps described above to train Doc2Vec models. The detailed methodologies in training Doc2Vec models and evaluating them are also described in the Appendix. Results from our experiments suggested that the Doc2Vec model trained with the PV-DM algorithm for 150 epochs and 50-dimensional document vectors was the most optimal model.

With this model, we were able to extract document vectors for any scam report not only those in our corpus but also unseen ones. Figure 6 shows a simple illustration of this using Document Tag ID #20, an actual scam report in the corpus. The 50-dimensional document vector produced by the trained Doc2Vec can be regarded as a numerical representation of the text in Document Tag ID #20. Put differently, the document vector is said to encapsulate the meaning of this document.

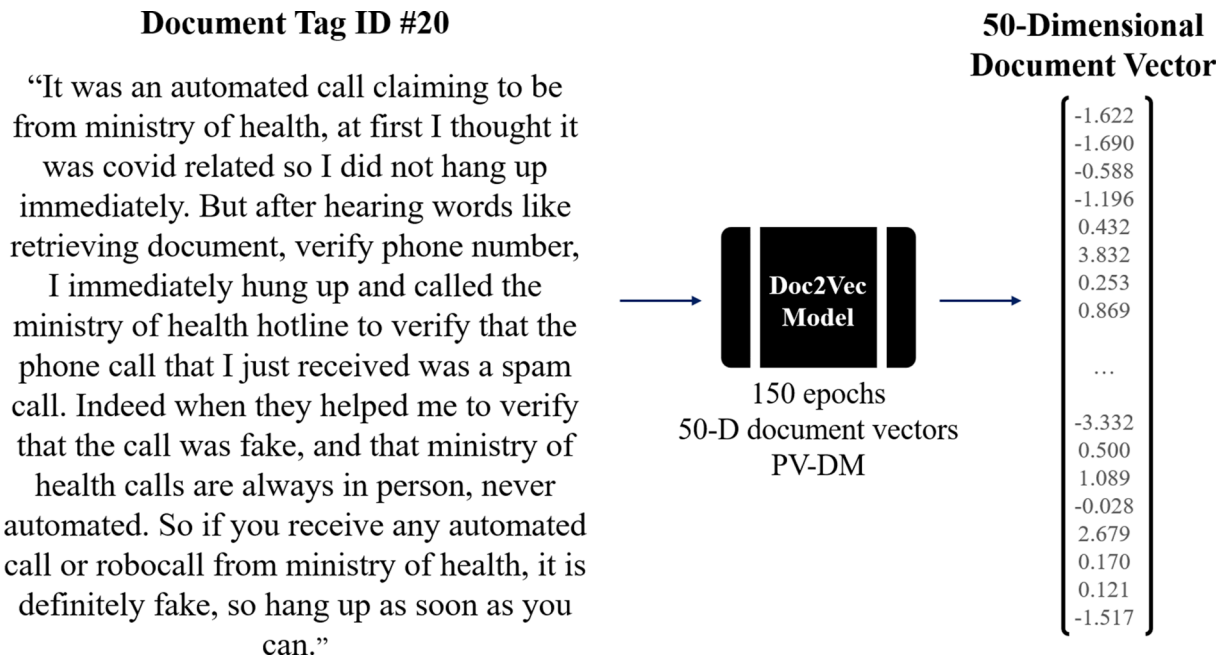

Fig. 6 Extracting document vectors of scam reports in the text corpus using trained Doc2Vec model

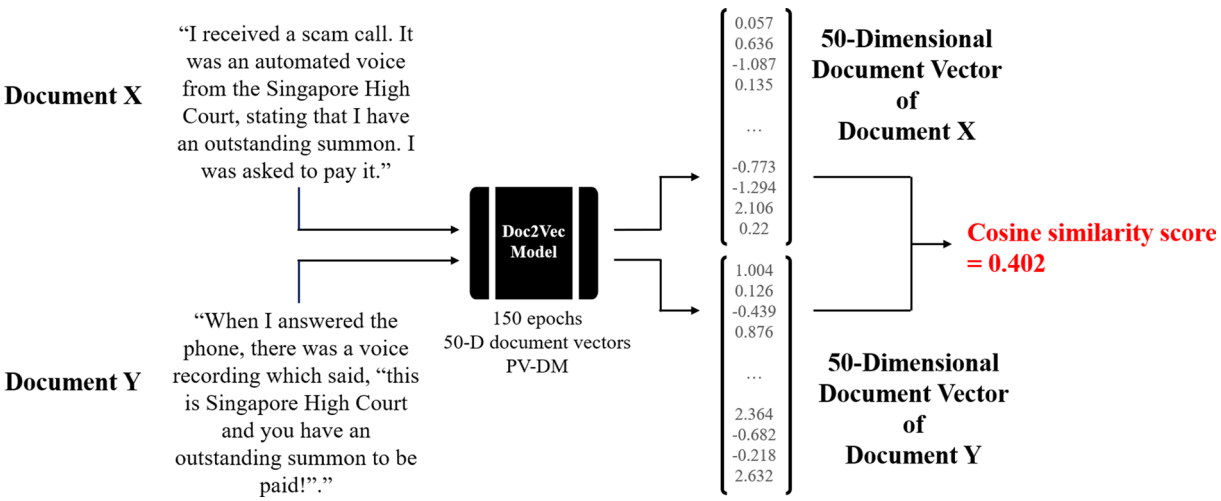

Fig. 7 Inferring document vectors of unseen scam reports, Documents X and Y, from trained Doc2Vec model

In practice, a more common scenario would be to apply the trained Doc2Vec model on new and unseen scam reports, in order to find potential links to similar scam reports from the existing corpus. The trained Doc2Vec model can be used to similarly infer document vectors for any unseen scam report. Similarity between these scam reports can then be quantified by measuring cosine similarity between their corresponding document vectors. Figure 7 depicts this process for the following two hypothetical scam reports—Document X and Document Y. Using the document vectors for both scam reports, the cosine similarity score is computed to be 0.402.

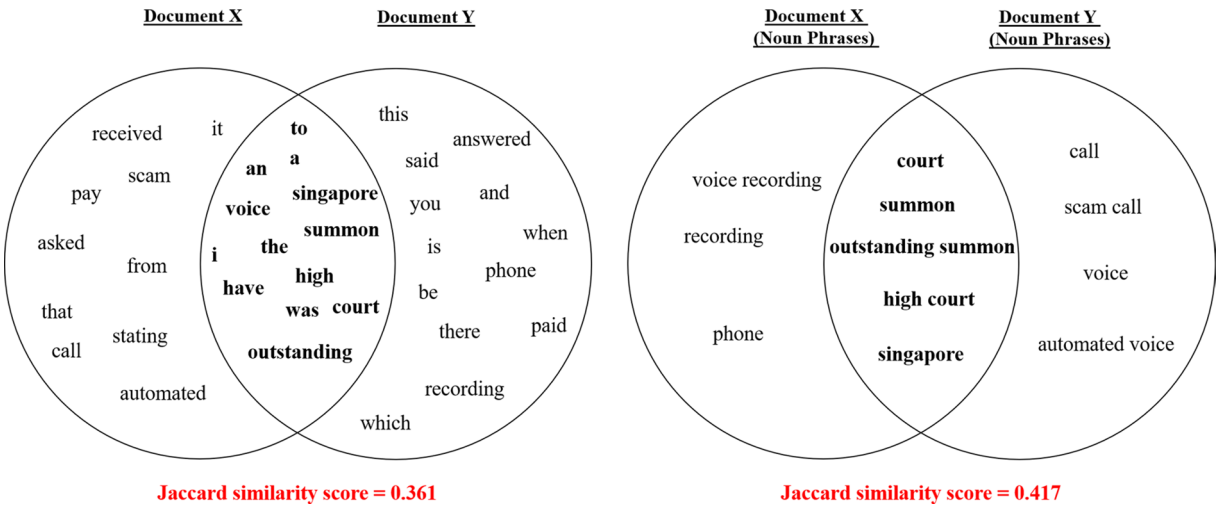

Fig. 8 Illustrations of Jaccard similarity using all words and only noun phrases

Document X	**"I received a scam call. It was an automated voice from the Singapore High Court, stating that I have an outstanding summon. I was asked to pay it."**
Document Y	"When I answered the phone, there was a voice recording which said, "this is Singapore High Court and you have an outstanding summon to be paid!"."

To retrieve scam reports in our corpus which are most similar to an unseen scam report, we first infer a document vector for the unseen scam report using the Doc2Vec model. Next, we compute cosine similarity scores between the inferred document vector and the document vectors of all scam reports in the corpus. The scam reports which are most similar to the unseen scam report by the vector-based approach would be those with the highest cosine similarity scores.

Text-based approach

The second method used in measuring similarity of modus operandi in scam reports is the text-based approach. In computing Jaccard similarity, each document is regarded as a set of words. Jaccard similarity measures the proportion of all words that are common to both documents. In our work, we applied a modified version of Jaccard similarity. We hypothesised that scam reports with similar modus operandi would make reference to similar noun phrases such as names of government organisations, banks and social media platforms. Therefore, instead of computing Jaccard similarity using all words, which is the default, we modified the computation to be based on noun phrases[3] only. Figure 8 contextualises this idea using Documents X and Y.

Figure 8 shows how Jaccard similarity between Documents X and Y would differ if we were to use all words compared with noun phrases only. Clearly, Documents X and Y had a higher Jaccard similarity score and were "more similar" when compared using noun phrases. Here, we argue that the modified Jaccard similarity provides a more accurate picture of the degree of similarity between two scam reports. In other words, the higher the Jaccard similarity score, the more noun phrases two scam reports have in common and the more similar their modus operandi is likely to be. In addition, text reduction techniques such as stemming and lemmatisation may further help in generating Jaccard similarity scores that better reflect how similar two scam reports are. For example, the words "summons" and "summon" can be treated as common since both can be reduced to the same base word "summon", thereby giving a higher Jaccard similarity score.

To find scam reports which were most similar in modus operandi to any input scam report, we first used the 'spaCy' library (Matthew, Montani, Van Landeghem, & Boyd, 2013) in Python to identify and extract noun phrases using two matching patterns: phrases that contain consecutive nouns with at least one noun; and phrases that start with one adjective followed by consecutive nouns with at least one noun. Thereafter, Jaccard similarity using noun phrases can be computed between the input scam report and every scam report in the text corpus. Scam reports with the highest Jaccard similarity scores would be deemed most similar to the input scam report in terms of modus operandi.

[3] A noun phrase is a phrase containing a noun and optionally other kinds of words such as pronouns and adjectives. Examples include "scam call", "internet connection", "local landline" and "police".

Generating key terms from similar scam reports

In the previous section, we highlighted the importance of retrieving and isolating scam reports with similar modus operandi in producing reliable crime scripts. More specifically, this prerequisite step seeks to ensure that the key terms contained within a set of similar scam reports reflect the characteristics of their modus operandi, which can be perceived as different components or stages in the scam script. Generating scripts that accurately convey the different stages involved in the scam will ultimately facilitate disruption and prevention efforts.

To illustrate, here we might consider internet love scams. Isolating a small sample of internet love scams with similar modus operandi may uncover key terms such as "dubai", "engineer", "transfer", "emergency" and "western union". We can then infer that stages of this scam script may include the scammer introducing himself to victims as an engineer from Dubai, the scammer mentioning about some emergency, and victims being asked to transfer money via Western Union, a global money remittance company. One possible intervention point in this script would be Western Union and measures could be taken to educate its employees to identify tell-tale signs of likely scam victims intending to remit money overseas, in the efforts to disrupt such scams.

To generate key terms from a set of similar scam reports, we used the scam reports whose text had been pre-processed (as described above) and removed stop words[4] therefrom using the 'nltk' library (Bird et al., 2009) in Python. Next, we applied TF-IDF to extract *n-grams*, or a consecutive sequence of *n* words. Besides extracting single words, known as *unigrams*, it was also possible to extract combinations of two words and three words, known as *bigrams* and *trigrams* respectively. Each n-gram was given a TF-IDF score according to its importance to the given set of scam reports. The higher the TF-IDF score, the more important the n-gram was.

Table 2 shows the resulting top n-grams by TF-IDF scores extracted from Documents X and Y using this approach. Although only two scam reports were examined, the n-grams generated gave useful information about the main features of this modus operandi.

Since script analysis of scams should convey characteristics of scams as a sequence of events, we took one step further to represent n-grams sequentially using a directed graph. For each n-gram, we computed its median index position across the set of similar scam reports. The n-grams were then sorted by their median index positions. Visualising n-grams as a directed graph provides insights on the different stages of the modus operandi and their characteristics, thereby greatly facilitating the scripting of scams. The case study highlighted in the next section will put these methodologies into context. In addition, given that our text corpus originated from different victims, there were understandably variations in the ways scam stories were described, even for a set of similar scam reports. Therefore, a significant and necessary assumption this approach makes is that the order in which victims describe their scam stories generally reflects the order in which the scams took place. While we acknowledge this is not ideal, we assert that it represents an appropriate first step, which is best assessed by subsequently analysing the sequences of actions such an approach will generate.

Table 2 Key terms extracted from Documents X and Y

N-grams	TF-IDF score
Court	0.3400
High court	0.3400
Singapore	0.3400
Singapore high	0.3400
Summon	0.3400
Answered phone	0.2458
Voice recording	0.2458
Automated	0.2321

Results: a case study of high court impersonation scams

Having provided an overview of our analytical workflow, we now demonstrate how it can be applied to help in script analysis of scams, using the example of High Court impersonation scams identified within our corpus. The scam in question is characterised by potential victims receiving a telephone call purportedly from the "Singapore High Court", with an automated recorded message in English and Mandarin. Potential victims are asked to press "9" before being directed to a person claiming to be a Court officer. On speaking to this supposed official, they are told to attend Court on the pretext of being involved in a crime or as a result of pending summons. The official would then request personal details from the potential victim such as names and identification numbers.

To illustrate how we can identify such scams that share similar modus operandi in our corpus and subsequently generate insights on the different stages involved, we will use the following hypothetical textual description as our input:

[4] Stop words are common words that contribute limited meaning in text. Examples include "to", "an", "the" and "which".

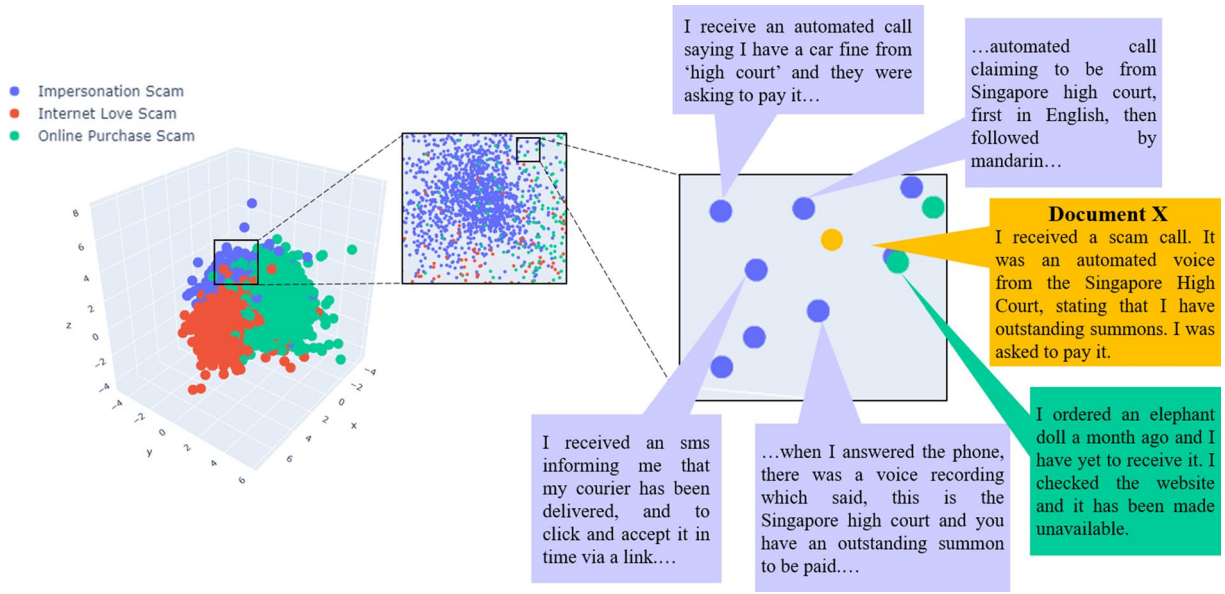

Fig. 9 Mapping of Document X's document vector on a 3D vector space

Document X "I received a scam call. It was an automated voice from the Singapore High Court, stating that I have an outstanding summon. I was asked to pay it."

The selected Doc2Vec model in the vector-based approach was used to infer 50-dimensional document vectors, not only for Document X but for all scam reports in the corpus. The left image in Fig. 9 shows a three-dimensional (3D) projection of document vectors belonging to the top three scam categories using Principal Components Analysis (PCA).[5] The inset in Fig. 9 illustrates the position of Document X's document vector on a 3D space, marked by the data point coloured in orange. The data points in its immediate neighbourhood can be regarded as document vectors of scam reports which were most similar to Document X by cosine similarity (as discussed above). In this illustration, several data points close to the orange data point were indeed High Court impersonation scams, but there were also others which were unrelated.

Using document vectors inferred by the Doc2Vec model, we computed cosine similarity scores between Document X and every scam report in the corpus. The scores were ranked to find scam reports that were most similar to Document X. The top eight most similar scam reports to Document X by cosine similarity are shown in Table 3.

Evidently, the vector-based approach did not yield good results. As Table 3 shows, most of the top eight most similar scam reports were of different modus operandi as Document X's. Only two scam reports—Tag IDs #972 and #1787—pertained to the High Court impersonation scam. It turned out that the text-based approach, in this case, was more effective in isolating scam reports most similar to Document X by modus operandi. The top eight scam reports most similar to Document X using the modified Jaccard similarity are presented in Table 4. All eight scam reports were highly similar to Document X in terms of the modality of the High Court impersonation scam. There were with several noun phrases in common such as "high court", "automated voice" and "outstanding summon" amongst the top scam reports. This demonstrates the effectiveness of using Jaccard similarity with noun phrases.

For this case study, the text-based approach proved to be more effective than the vector-based approach. However, this may not be the case for other scam reports. The choice between these two approaches would depend greatly on the nature of scams and how they were described by victims. In our work, it was observed that text-based approach tended to be more effective for scams which were relatively more predictable. By this we mean that the words and phrases used by victims to describe their experiences, as well as the order in which they were mentioned in the reports, were more likely to be similar.

The text-based approach, however, appeared to work less effectively for scams whose modus operandi were

[5] PCA is a method for reducing dimensionality of high-dimensional data. It transforms a large set of features into a smaller set that retains the most valuable information.

Table 3 Top eight most similar scam reports using vector-based approach

Rank	Tag ID	Cosine similarity Score	Scam report (pre-processed)	Scam category
1	972	0.669	call from automated voice message stating that he is from the high court saying that I have missed submitting an important document and asked to press to ask more questions	Impersonation scam
2	4339	0.660	i simply ignore and hung up the call	Impersonation scam
3	1219	0.634	the dhl scam is back received a call that started off with an automated voice message in mandarin, informing me that I have a parcel from dhl. I was asked to press to speak to an operator. Hung up the call as i knew it was a scam. Call came from this number which might be a spoofed number	Impersonation scam
4	55	0.627	i received a scam spam call from this number asking for details about my singtel internet connection. I am not even a subscriber of any singnet singtel services. so, I ended the call. I thought it would be great to share my experience here to warn others of this scam	Phishing scam
5	2484	0.623	i received a call from on th may. It was a female voice claiming that it was a call from singapore police headquarters. I hung up immediately as I sensed something was wrong	Impersonation scam
6	1787	0.621	got a call aug at. private message from this no advising I have outstanding summons not cleared	Impersonation scam
7	2422	0.619	received this scam call from the singapore police force this morning	Impersonation scam
8	825	0.601	i received a call from an unknown number. Heard an automated voice message informing me that i have an unclaimed package. It spoke in english, then mandarin. I hung up immediately	Impersonation scam

Table 4 Top eight most similar scam reports using text-based approach

Rank	Tag ID	Jaccard similarity score	Scam report (pre-processed)	Scam category
1	1866	0.500	automated robot voice called me and said it is from singapore high court, and i have outstanding summon also repeated in mandarin then i hung up. not sure what is the intention of the call	Impersonation scam
2	356	0.462	received a call from at. am on april. it was an automated voice stating the call is from singapore high court. put down the phone immediately after, so did not hear the rest of the message	Impersonation scam
3	279	0.400	i got the above call and it says i got an outstanding summon and to dial nine to talk to someone. someone answered and says its singapore high court the person sound like a local singaporean	Impersonation scam
4	141	0.400	got a call from a local singapore number. answered the call and it was an automated voice in english saying the high court of singapore was serving a summons on me. i immediately hung up. this is a total scam, the high court of singapore will never serve summons like this	Impersonation scam
5	1874	0.385	received a phone call with a voice message saying this is singapore high court. you have an outstanding summon. press	Impersonation scam
6	1803	0.385	call with automated voice saying i have summons from singapore high court. press to proceed. someone picked up the call but then did not speak anything	Impersonation scam
7	1026	0.375	i received a call from initially it was automated voice saying you have been summoned by singapore high court to get a document, for more details press. as i just received a dhl scam call yesterday, i did not proceed any further	Impersonation scam
8	1840	0.375	received a call from this morning. a computer voice said this is the singapore high court. you have a summon pending. please press after the beep for more information. i became suspicious and cut the call. beware people	Impersonation scam

more varied across reports. For example, exploratory analyses of internet love scams showed that scammers would use different names on different online dating platforms and present a more diverse range of stories. Consequently, there would be a greater variety in the words and phrases used by victims to describe such scams. Thus, the text-based approach would be less effective. Instead, the vector-based approach might be a better choice in identifying scams with contextually similar modus operandi. Notwithstanding, this observation was based on limited exploratory analysis of differing scam types – and warrants further investigation in subsequent studies. For instance, it is also reasonable to investigate how cosine and Jaccard similarity scores are distributed for different types of scams and understand whether there could be similar documents with low similarity scores and why. Nevertheless, it is clear that both approaches have their relative merits. Ultimately, given our aim to support human analysts, it would seem sensible that in analysing a given series of reports, an analyst might explore the

Table 5 Top 10 n-grams and their respective TF-IDF scores

Unigrams	TF-IDF Score	Bigrams	TF-IDF Score	Trigrams	TF-IDF Score
call	3.757	singapore high	2.132	singapore high court	2.002
singapore	2.804	high court	2.107	court outstanding summon	1.442
court	2.739	outstanding summon	1.612	high court outstanding	1.442
received	2.731	phone call	1.488	received phone call	1.371
high	2.688	court outstanding	1.468	outstanding summon press	1.319
press	2.617	received phone	1.410	call voice message	1.257
voice	2.567	message saying	1.370	message saying singapore	1.257
saying	2.278	summon press	1.353	phone call voice	1.257
summon	2.184	call voice	1.286	saying singapore high	1.257
outstanding	2.112	saying singapore	1.286	voice message saying	1.257

effectiveness of both approaches and establish which is likely to be most effective given their own data.

Having determined that the text-based approach was the most effective in the context of our case study of High Court impersonation scams, we next extracted unique words and phrases from those reports deemed similar. We examined the top 0.5%[6] of scam reports most similar to Document X (that is, 23 scam reports). While the selection of this threshold is wholly arbitrary, it was chosen here to concentrate analysis on a relatively small number of scam reports that were very similar to Document X, without introducing too much noise in the form of scam reports with deviant modus operandi. From these scam reports, we extracted a total of 150 unigrams, 288 bigrams and 326 trigrams. The top 10 unigrams, bigrams and trigrams ranked by TF-IDF scores are shown in Table 5.

Table 5 highlights important n-grams which are characteristic of the High Court impersonation scams. However, it does not adequately inform about how this scam typically unfolded. To improve the way such key terms are presented, we further harnessed information in the set of similar scam reports by taking note of the index position corresponding to a particular n-gram in each scam report. For example, the index position of the bigram "singapore high" in Tag ID #1866 was 52, which meant that "singapore high" appeared at the 52th index position of this document. The n-grams were then arranged by their median index positions across the set of similar scam reports. Table 6 illustrates this idea for the top 20 unigrams amongst the top 23 scam reports most similar to Document X. As discussed previously, this approach makes a significant and necessary assumption that the order in which victims describe their scam stories generally reflects the order in which the scams took place. While this may not always be the case, we believe in the absence of better sequential insights these characteristics should not be lost. Moreover, the implications of such an assumption can to some degree be evaluated by assessing the plausibility of generated sequences relative to a given scam report.

Table 6 Top 20 unigrams sorted by median index position

Unigrams	TF-IDF Score	Median Index Position
received	2.731	0.0
automated	1.732	9.0
phone	2.076	9.0
call	3.757	11.5
number	1.281	14.0
voice	2.567	26.5
message	2.073	30.5
saying	2.278	35.0
singapore	2.804	41.0
high	2.688	52.5
court	2.739	57.5
english	1.253	60.5
outstanding	2.112	63.0
summon	2.184	74.5

When arranged by median index positions, the sequence of unigrams does indeed provide a better intuition about how the scam took place, at least from the victims' perspectives. The same information presented in Table 6 can be visualised as a directed graph, as shown in Fig. 10. This directed graph consists of a series of nodes, where each node represents an n-gram.

[6] In practice, this threshold should vary, depending on the nature of the scam in question. On one hand, too stringent a threshold would greatly limit the amount of information we can harness insights from. On the other hand, too liberal a threshold would introduce noise, and consequently affect the quality of model results. Therefore, some experimentation would be recommended to determine an appropriate threshold.

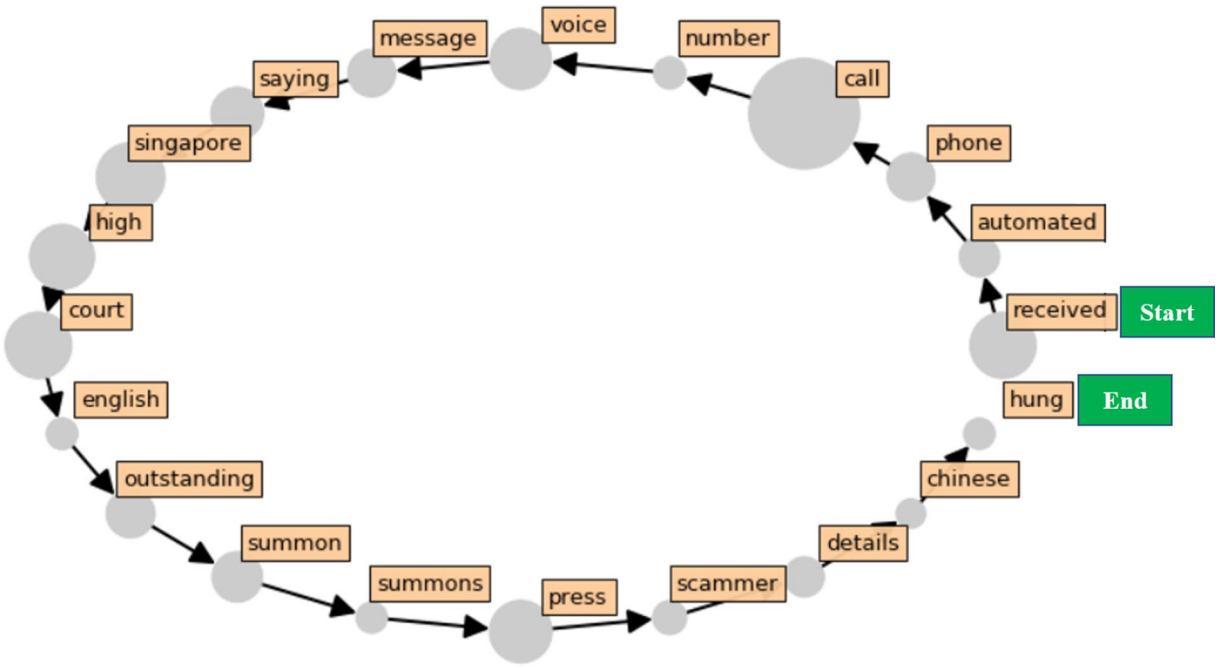

Fig. 10 A directed graph showing the top 20 unigrams in sequence

The size of the node reflects the TF-IDF score of the corresponding n-gram and its relative uniqueness amongst all n-grams in the set of the 23 most similar reports. The bigger the node, the higher the TF-IDF score, the more unique the n-gram was to those reports. The assumption is that n-grams with higher TF-IDF scores and therefore more unique tend to better define the characteristics of these reports. The directionality of the graph conveys the sequence in which the scams unfolded. Besides generating only one type of n-gram from a set of similar reports, the same could also be done for a combination of unigrams, bigrams or trigrams.

Extracting key terms using TF-IDF allowed us to identify common characteristics of the High Court impersonation scam. Moreover, by arranging them using their median index positions and visualising them as a directed graph, we were able to better understand the sequence of events involved in the scam from victims' points of view. Assuming that one had little or no knowledge about the High Court impersonation scam, the directed graph in Fig. 10 would imply the following stages:

1. Victims receive a phone call;
2. Victims hear an automated voice message;
3. The automated voice claims to be from the Singapore High Court;
4. The automated voice mentions "outstanding summons";
5. Victims are asked to press a number;
6. Victims are directed to a scammer speaking in Chinese; and
7. The scammer asks for victims' details.

These stages were inferred entirely from the directed graph and would be insightful in the script analysis in the High Court impersonation scam. Importantly, not all stages or characteristics needed to be mentioned in individual scam reports. The approach taken is capable of leveraging the collective wisdom of multiple similar reports to identify a consensus in terms of likely crime event sequences. Therefore, we were able to derive insights that were not originally present in Document X. For instance, from the directed graph, we would expect that at some point during the phone call, the victim would be asked to press a number and be directed to a Chinese-speaking person, who would then ask for victims' details. These information were not present in the textual description of Document X and would have been missed if we were to analyse only Document X.

That said, it is important to highlight that our analytical approach, as demonstrated in this case study, does not automatically generate scripts per se. Instead, it produces a chronological sequence of key terms which we hypothesise capture the key procedural elements of a particular modus operandi. To transform the output of our approach into actual meaningful scripts, much

interpretation, domain knowledge and human judgement is necessary. In other words, our workflow helps facilitate crime analysts in the development of scripts through the automatic sense-making of collections of textual documents.

Discussion

In this paper, we described novel research efforts that sought to explore how NLP methods might be used to support crime script analyses, in particular focusing on victim scripts, and providing a case study using data describing scams posted by the public in Singapore using the 'Scam Alert' website. Analyses of these data encompassed several interconnected phases. First, we identified methods for collating similar scam reports, in this case, applying cosine similarity scores to document vectors inferred by a Doc2Vec model in combination with a modified Jaccard similarity metric that sought to identify reports containing common noun phrases. Second, we explored how key terms could be extracted from these sets of similar scam reports to identify key elements and actions associated with a particular scam. Finally, analysing these terms, we demonstrated a simple method for deriving chronological insights associated with the different procedural stages of a given scam.

This workflow, we argue, has the potential to provide significant benefits to those organisations or individuals who seek to conduct crime script analyses. The procedural analyses of crime through crime scripts requires a significant amount of human effort and expertise. Large quantities of administrative data are typically collated and subsequently need to be analysed. This can involve analysts reading lots of textual descriptions of offences, with the aim of identifying the key elements and actions associated with the commission of a particular offence. In addition to being highly resource-intensive, this process is also likely subject to a range of biases (Birks et al., 2020). The case study presented above demonstrates that NLP has the potential of significantly alleviating resource requirements associated with crime script analyses. Insights that can take a considerable number of person hours to extract (and thus may be rarely extracted due to resource pressures) can potentially be obtained in a much shorter time with the use of NLP. While such approaches will only ever be useful in supporting essential human analyses, we are optimistic about the potential for methods such as those described here in helping those who seek to deliver better understandings of crime problems, and in turn more targeted interventions that seek to reduce crime.

A second point of note here relates to the primary data source used in this study, namely victim self- reports of scams available through the Singapore-based 'Scam Alert' website. To date, most research concerning the application of NLP methods to crime or crime-related phenomena have, for a range of understandable reasons, relied on researchers gaining access to protected data through formalised data sharing agreements. While such protocols are clearly necessary for a host of reasons, one might well argue that a general paucity of readily accessible crime event related textual data is likely to reduce the speed of innovation in the application of NLP to various elements of the crime analysis endeavour. As such, again we are optimistic about the potential insights that might be derived from the ever-increasing numbers of public datasets available to the research community, such as the one analysed here.

That said, the approach presented in this article still has several limitations. The first concerns script quality. While our approach provides an efficient means towards generating scripts, it does not guarantee the quality of the resulting scripts, which is influenced by how data is collected (Borrion, Dehghanniri, & Li, 2017). Since our data originated from different victims, there were huge linguistic variations. These inconsistencies introduce noise, which undoubtedly affect the performance of NLP models. What we did to mitigate this was to iteratively identify inconsistencies and rectify them manually during data cleaning and text pre-processing. However, this approach is not absolute and is unlikely to be feasible at scale. Though evaluating quality of scripts generated using NLP methods is beyond the scope of this research, there is nonetheless value to doing so. Using the list of twelve quality criteria for crime scripts presented by Borrion (2013) seems a good starting point. Furthermore, while we have made certain assumptions about the relative effectiveness of the vector-based and text-based approaches to finding similar scam reports, they have yet to be rigorously tested. Thus, it also seems a reasonable extension of this study to evaluate how well this workflow can generalise to other scam reports in the corpus or perhaps even textual data from a completely different crime domain.

A second limitation concerns the specific use of TF-IDF and its effects on the results. The effectiveness of TF-IDF is only as good as its input data. The assumption we made was that n-grams with higher TF-IDF scores are more unique and important in defining the characteristics of a collection of scam reports. This depends on how similar the scam reports are to one another. If the input scam reports are hugely varied in terms of their modus operandi, the top n-grams generated by TF-IDF may not be so informative. Moreover, even if the input scam reports share very similar modus operandi, such as in the case study discussed in this paper, it is also possible that some n-grams may be less useful than others in furthering our

understanding of the type of scam inherent amongst those reports. For example, the bigram, "saying singapore" was amongst the top 10 bigrams by TF-IDF score in Table 5 but it does not provide as much information about the scam as other bigrams, such as "high court" or "outstanding summon". There is also a likelihood that an n-gram, which is relevant in defining the characteristics of a scam, is not given a high TF-IDF score. This could happen if the n-gram appears commonly in several scam reports, thereby undermining its 'uniqueness'. After all, by the definition of TF-IDF, the more documents a particular n-gram appears in, the lower the TF-IDF of that n-gram. Given these intricacies, there is a degree of uncertainty about the quality of the results arising from the use of TF-IDF and subsequent work should seek to assess the generalisability of this method to other scam types as well as the practical utility it provides to crime analysts conducting crime script analyses.

The third limitation relates to our methodology of evaluating Doc2Vec models. To do so, we developed a novel metric called *Similarity-Dissimilarity Quotient* (SDQ) that measures how well document embeddings inferred by a Doc2Vec model can differentiate between similar and dissimilar scam reports. This approach is described in further details in the Appendix. We manually identified candidate sets of similar and dissimilar scam reports from our text corpus to be used as benchmarks in evaluating Doc2Vec models using SDQ. This process was at best subjective. An alternative, and likely more effective, method for identifying sets of similar and dissimilar scam reports would be to first train a Doc2Vec model on the text corpus, and then use the trained model to find the second-most similar and the least similar scam reports by cosine similarity for each of n scam reports randomly selected from the corpus. Unlike the approach taken in this research, this alternative method would provide a quantitative basis for selecting candidate scam reports, which could translate to lesser ambiguity for Doc2Vec models in recognising similar and dissimilar scam reports.

Moving forward, there are several areas of future research which we believe will support the application of NLP in script analysis. In spite of our initial assessment of Doc2Vec as an ideal method for encoding scam reports as document vectors, we saw briefly through the case study that it could have its weaknesses in identifying scam reports with similar modus operandi. While more research should be carried out to evaluate the relative merits of the Doc2Vec and text-based methods for different scam types, it is also recommended that alternative NLP techniques to encode textual data as document vectors be explored. These include averaging Word2Vec embeddings of all words, using TF-IDF, training deep neural networks as well as using pre-trained transformer models like BERT. A second possible extension is to train Doc2Vec models on text without stop words. Here, we did not remove stop words from the text corpus used to train Doc2Vec models, because it was assessed that stop words played important roles in the grammatical structures of scam reports and thus would contribute towards good document vectors. Nevertheless, there is scope to further investigate the quality of document vectors from Doc2Vec models trained on a corpus without stop words. Lastly, apart from using TF-IDF to extract key words and phrases from a collection of similar scam reports, we could also explore topic modelling techniques such as Latent Semantic Analysis and Latent Dirichlet Allocation. These techniques could be used to extract key topics inherent within those scam reports. Put differently, topic modelling could help scripters "see" the bigger picture, as well as better understand the intricacies behind the modus operandi of a particular scam.

Conclusion

While the methods presented in this paper were exploratory and limited examples were highlighted, we believe they underscore further potential for NLP methods in harnessing hidden insights from crime related free text data. With this in mind, we hope our work can provide a starting point for further research into using NLP for script analysis. There is no reason to believe that similar methodologies could not be applied to other types of scams or crimes in general. Relative to manual analyses of unstructured information, as is typically the case in traditional script analysis, NLP could make the script analysis process more efficient, especially where unstructured secondary free text data are available to analysts. NLP could even help pave the way towards the development of more systematic crime scripting methods, as advocated strongly for within the research community (Borrion, Dehghanniri, & Li, 2017; Borrion, 2013).

As with previous exploratory applications of NLP to the analyses of crime problems, it is clear that data analytic solutions alone cannot, and should not aspire to, replace the judgement of domain experts. Nevertheless, they may generate considerable added value from textual administrative data that is routinely collected by police agencies but rarely exploited in ways that commensurate with the investment associated with their capture. By harnessing NLP to carry some of the analytical burden, data science may offer viable means to more readily and rapidly support crime script analyses, in turn increasing the likelihood of identifying viable points of intervention, and ultimately supporting those who seek to prevent crime and its diverse associated harms.

Appendix

Training and evaluating Doc2Vec models

This Appendix describes our methodology in training and evaluating Doc2Vec models, as well as the results of our experiments.

Training Doc2Vec models

The training of Doc2Vec models involved a two-staged process. In the first stage, the Doc2Vec models were trained for 10, 25, 50 and 100 epochs. We also varied the dimensionality of document vectors—20, 30, 40 and 50 dimensions. For each combination, we trained both PV-DM and PV-DBOW algorithms. From the first stage, we determined the optimal dimension size of document vectors and the superior Doc2Vec algorithm, before proceeding to vary only the number of epochs in the second stage over a wider range. We experimented with training Doc2Vec models for 25, 50, 75, 100, 150, 200, 250, 300, 400 and 500 epochs. The optimal amount of training was then determined by monitoring the number of epochs it took before a model's evaluation performance started to decline (Mohr, personal communication, n.d.).

Evaluating Doc2Vec models

A Doc2Vec model's performance was measured using a novel framework we developed, known as the *Similarity-Dissimilarity Quotient* (SDQ). Consider a set of three candidate documents shown in Fig. 11, where Documents A and B are similar, but they are both dissimilar to Document C. In this paper, we refer to a set of three candidate documents with these properties as a *triplet*. The first step in evaluating Doc2Vec models using SDQ was to manually select eight triplets from our text corpus.

For each trained Doc2Vec model, we inferred document vectors for the three scam reports in each triplet. Using these document vectors, we computed cosine similarity scores between one another. Following this, we transformed all cosine similarity scores to a range between 0 and 1 using the sigmoid function. Next, we computed absolute SDQ, defined by the following equation:

$$\mathrm{SDQ}_{\mathrm{abs}} = \frac{\sigma(\cos\theta_{\mathrm{AB}})}{\sigma(\cos\theta_{\mathrm{BC}}) \times \sigma(\cos\theta_{\mathrm{AC}})}$$

where cos θ_{AB} is the cosine similarity between Documents A and B, cos θ_{BC} is the cosine similarity between Documents B and C, and cos θ_{AC} is the cosine similarity between Documents A and C. The more similar Documents A and B are, the bigger the numerator is expected to be. Conversely, the more dissimilar Documents A and B each are to Document C, the smaller the denominator should be. Given that cos θ ranges between -1 and 1, the maximum and minimum values of $\mathrm{SDQ}_{\mathrm{abs}}$ are

$\mathrm{SDQ}_{\mathrm{max}} = \frac{\sigma(1)}{\sigma(-1)\times\sigma(-1)} \approx 10.107$ and $\mathrm{SDQ}_{\mathrm{min}} = \frac{\sigma(-1)}{\sigma(1)\times\sigma(1)} \approx 0.503$.

Therefore, we normalise the absolute SDQ by the following expression,

$\mathrm{SDQ}_{\mathrm{norm}} = \frac{\mathrm{SDQ}_{\mathrm{abs}} - \mathrm{SDQ}_{\mathrm{min}}}{\mathrm{SDQ}_{\mathrm{max}} - \mathrm{SDQ}_{\mathrm{min}}}$,

which ranges between 0 and 1. The higher the normalised SDQ, the better a Doc2Vec model is in inferring document vectors capable of recognising similar and dissimilar documents in a triplet. The process is repeated for all eight triplets before the mean normalised SDQ score was taken. This process is illustrated in Fig. 12. The Doc2Vec model that produced the highest mean normalised SDQ score would be the most optimal model.

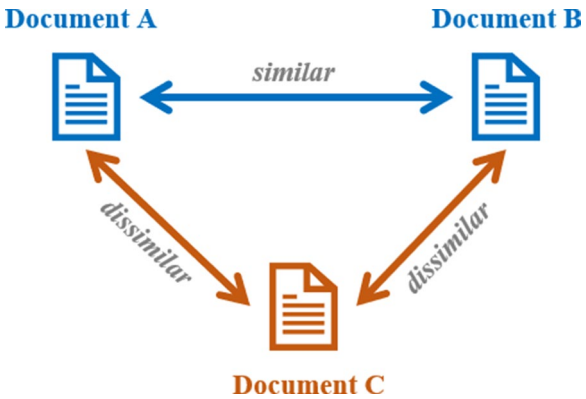

Fig. 11 An illustration of a triplet consisting of three candidate documents

Results of Doc2Vec model training

The results of tuning each hyperparameter in the first stage of model training are summarised by the boxplots in Figs.13, 14, 15 . From Fig. 13, PV-DM was notably superior over PV-DBOW, with a median normalised SDQ of 0.167 compared to a median normalised SDQ of 0.157 for the latter. For dimension size, Fig. 14 shows that Doc2Vec models with document vectors of 30 and 50 dimensions had median normalised SDQ of 0.166, outperforming those with document vectors of 20 and 40 dimensions. In addition, Fig. 15 suggests strong evidence that Doc2Vec models that were trained longer achieved higher normalised SDQ.

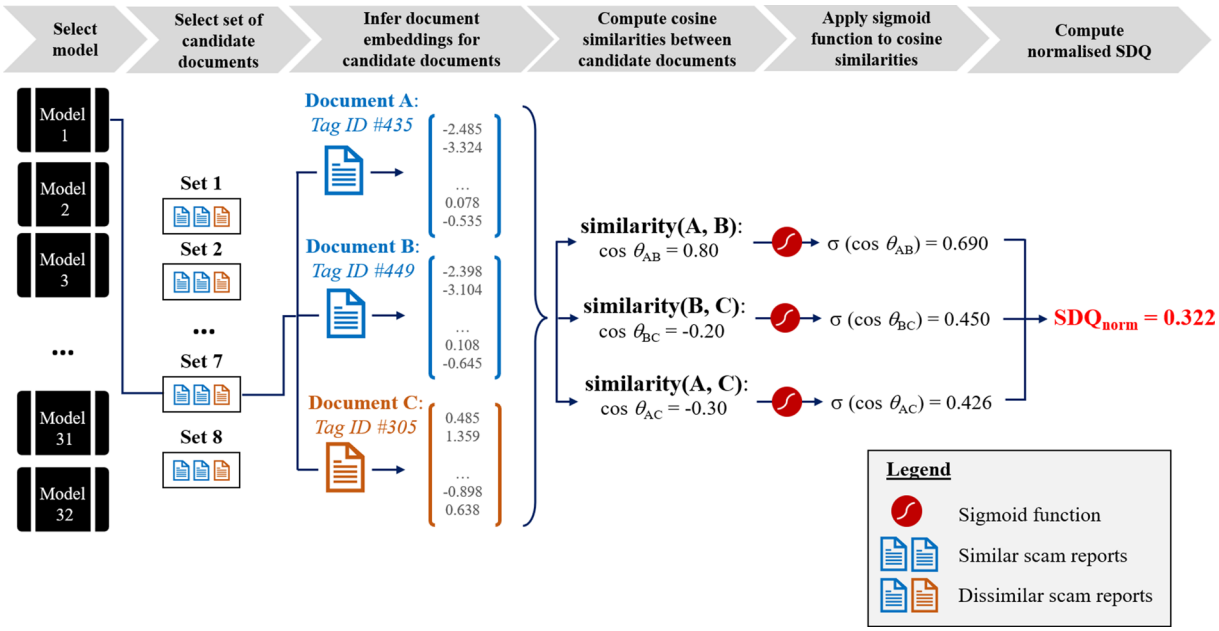

Fig. 12 An illustration of the process of computing SDQ

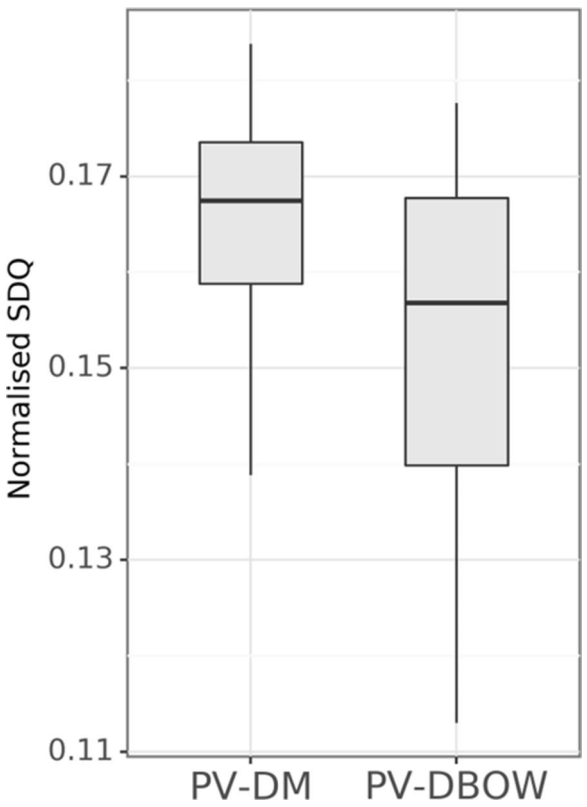

Fig. 13 Performance of 32 trained Doc2Vec models aggregated by type of Doc2Vec algorithm

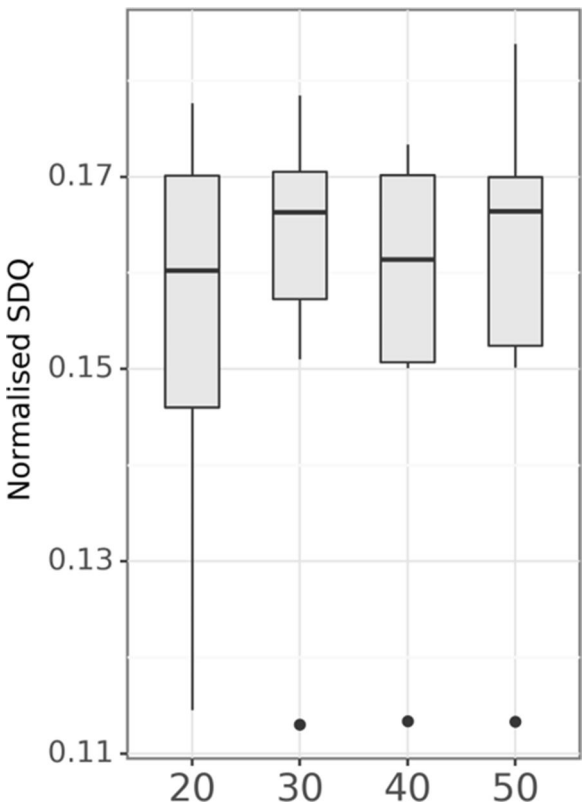

Fig. 14 Performance of 32 trained Doc2Vec models aggregated by dimension size

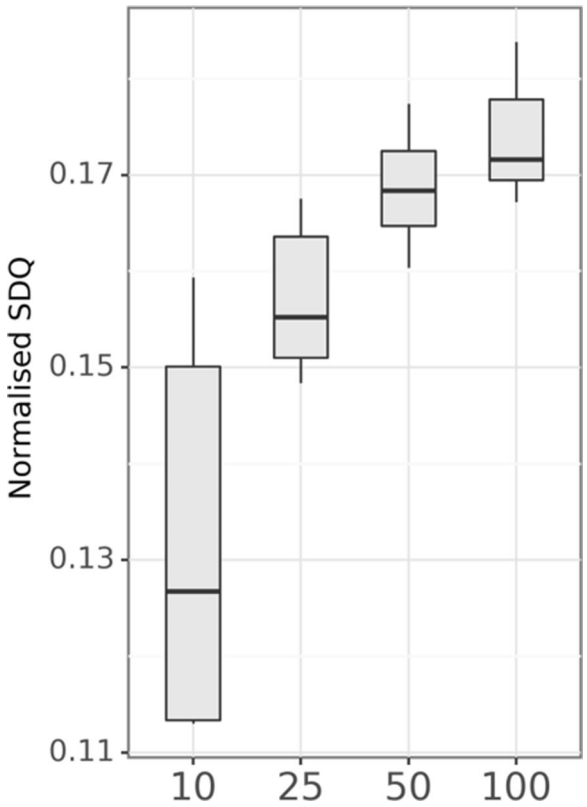

Fig. 15 Performance of 32 trained Doc2Vec models aggregated by number of epochs

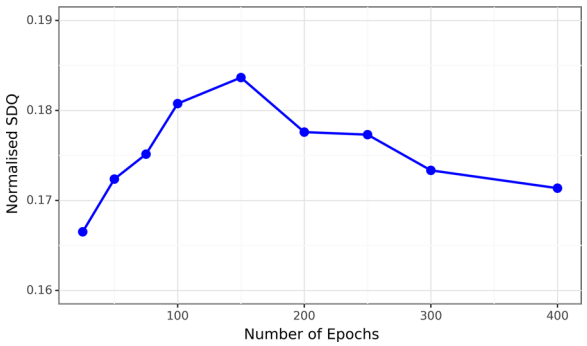

Fig. 16 Effect of number of epochs on Doc2Vec model performance

In the second stage, we used the PV-DM algorithm and kept dimension size of document vectors fixed at 50 while varying only the number of epochs of training over a wider range. Figure 16 shows that normalised SDQ was the most optimal at 150 epochs of training, beyond which the performance started to decline.

Given these results, it seemed a reasonable conclusion that the Doc2Vec model trained with 150 epochs, PV-DM algorithm and 50-dimensional document vector was the most optimal.

Acknowledgements
The authors would like to extend their gratitude to the National Crime Prevention Council (NCPC) of Singapore for approving the use of data from ScamAlert.sg, which formed the basis of this research.

Author contributions
ZLT designed the study, analysed the data, applied various methodologies and authored the paper. DB advised on the research direction and methodologies and co-authored the paper. Both authors read and approved the final manuscript.

Funding
This work was supported by Wave 1 of The UKRI Strategic Priorities Fund under the EPSRC Grant EP/T001569/1, particularly the "Criminal Justice System" theme within that grant and The Alan Turing Institute'.

Availability of data and materials
Source code for the project can be found at the following GitHub repository: https://github.com/zeyalt/Crime-Script-Analysis-NLP

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 1 February 2022 Accepted: 12 November 2022
Published online: 02 February 2023

References

Al-Zaidy, R., Fung, B. C., Youssef, A. M., & Fortin, F. (2012). Mining criminal networks from unstructured text documents. *Digital Investigation, 8*, 247–160.
Bird, S., Loper, E., & Klein, E. (2009). *Natural Language Processing with Python*. Sebastopol: O'Reilly Media Inc.
Birks, D., Coleman, A., & Jackson, D. (2020). Unsupervised identification of crime problems from police free-text data. *Crime Science*. https://doi.org/10.1186/s40163-020-00127-4
Borrion, H., Dehghanniri, H., & Li, Y. (2017). Comparative Analysis of Crime Scripts: One CCTV Footage - Twenty-One Scripts. *European Intelligence and Security Infomatics Conference* (pp. 115–122). IEEE.
Borrion, H. (2013). Quality assurance in crime scripting. *Crime Science*. https://doi.org/10.1186/2193-7680-2-6
Brayley, H., Cockbain, E., & Laycock, G. (2011). The value of crime scripting: deconstructing internal child sex trafficking. *Policing A Journal of Policy and Practice, 5*(2), 132–143.
Buchanan, T., & Whitty, M. T. (2014a). The online dating romance scam: causes and consequences of victimhood. *Psychology Crime & Law, 20*(3), 261–283.
Chainey, S. P., & Berbotto, A. A. (2021). A structured methodical process for populating a crime script of organized crime activity using OSINT. *Trends in Organized Crime*, 273–300
Chiu, Y.-N., Leclerc, B., & Townsley, M. (2011). Crime script analysis of drug manufacturing in clandestine laboratories. *The British Journal of Criminology, 51*(2), 355–374.
Choi, K., Lee, J.-L., & Chun, Y.-T. (2017). Voice phishing fraud and its modus operandi. *Security Journal, 30*, 454–466.
Cohen, L. E., & Felson, M. (1979). Social change and crime rate trends: a routine activity approach. *American Sociological Review, 44*, 588–608.
Cornish, D. (1994). Crimes as scripts. *Proceedings of the International Seminar on Environmental Criminology and Crime Analysis*. Tallahassee: Florida Statistical Analysis Center

Cornish, D. B., & Clarke, R. V. (1986). *The Reasoning Criminal: Rational Choice Perspectives on Offending*. New York: Transaction Publishers.
de Bie, J. L., de Poot, C. J., & van der Leun, J. P. (2015). Shifting modus operandi of jihadist foreign fighters from the netherlands between 2000 and 2013: a crime script analysis. *Terrorism and Political Violence, 24*(3), 416–440.
Dehghanniri, H., & Borrion, H. (2021). Crime scripting: a systematic review. *European Journal of Criminology, 18*, 504–525.
Drew, J., & Moore, T. (2014). Automatic Identification of Replicated Criminal Websites Using Combined Clustering. *2014 IEEE Security and Privacy Workshops*, (pp. 116–123).
Ekblom, P., & Gill, M. (2016). Rewriting the Script: Cross-Disciplinary Exploration and Conceptual Consolidation of the Procedural Analysis of Crime. *European Journal of Criminal Policy and Research, 22*, 319–339.
Elyezjy, N. T., & Elhaless, A. M. (2015). Investigating Crimes using Test Mining & Network Analysis. *International Journal of Computer Applications, 126*(8)
Fischer, P., Lea, S. E., & Evans, K. M. (2013). Why do individuals respond to fraudulent scam communications and lose money? The psychological determinants of scam compliance. *Journal of Applied Social Psychology, 43*(10), 2060–2072.
Friedman, D. A. (2020). Imposter Scams. *Social Science Research Network Electronic Journal*
Graham, R., & Triplett, R. (2017). Capable guardians in the digital environment: the role of digital literacy in reducing phishing victimization. *Deviant Behavior, 38*(12), 1371–1382.
Hamisu, M., & Mansour, A. (2020). Detecting Advance Fee Fraud Using NLP Bag of Word Model. *2020 IEEE 2nd International Conference on Cyberspace (Cyber Nigeria)*, (pp. 94–97)
Kuang, D., Brantingham, J. P., & Bertozzi, A. L. (2017). Crime Topic Modeling. *Crime Science, 6*(1), 12.
Le, Q., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. *Proceedings of the 31st International Conference on Machine Learning*
Leclerc, B. (2014). Cognition and Crime: Offender Decision Making and Script Analyses. In *New developments in script analysis for situational crime prevention: Moving beyond offender scripts* (pp. 221-236). Abingdon: Routledge
Lin, C. (2022, February 16). *Channel News Asia*. Retrieved from Spike in scams drives up Singapore's overall crime levels in 2021: https://www.channelnewsasia.com/singapore/crime-levels-scams-rise-2021-2501736
Luo, X., Zhang, W., Burd, S., & Seazzu, A. (2013). Investigating phishing victimization with the heuristic systematic model: a theoretical framework and an exploration. *Computers & Security, 38*, 28–38.
Mbaziira, A., & Jones, J. (2016). A Text-Based Deception Detection Model for Cybercrime. *International Conference on Technology and Management*
Matthew, H., Montani, I., Van Landeghem, S., & Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python. Zenodo
Mbaziira, A., Abozinadah, E., & Jones, J. H. (2015). Evaluating classifiers in detecting 419 scams in bilingual cybercriminal communities. *International Journal of Computer Science and Information Security, 13*(7), 1–7.
Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *International Conference on Learning Representations*
Mohr, G. (n.d.). *Avoiding over-fitting in Doc2Vec (personal communication)*. Retrieved from https://groups.google.com/g/gensim/c/JtUhgUjx4YI/m/3tvXgnSgBgAJ
Naudé, M., Adebayo, K. J., & Nanda, R. (2022). A machine learning approach to detecting fraudulent job types. *AI & Society*
Nguyen, T. V. (2021). The modus operandi of transnational computer fraud: a crime script analysis in Vietnam. *Trends in Organized Crime*, 226–247
Norvig, P. (2018). Pyspellchecker
Osborne, J. R., & Capellan, J. A. (2016). Examining active shooter events through the rational choice perspective and crime script analysis. *Security Journal, 30*, 880–902.
Pattinson, M., Jerram, C., Parsons, K., Mccormac, A., & Butavicius, M. (2011). Managing Phishing Emails: A Scenario-Based Experiment. *Proceedings of the Fifth International Symposium on Human Aspects of Information Security & Assurance*
Phillips, R., & Wilder, H. (2020). Tracing Cryptocurrency Scams: Clustering Replicated Advance-Fee and Phishing Websites. *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, (pp. 1–8)
Poh, A. National Crime Prevention Council, personal communication (June 17, 2020)
Richardson, L. (2007). Beautiful soup documentation
Schraagen, M., Testerink, B., Odekerken, D., & Bex, F. (2018). Argumentation-driven information extraction for online crime reports. *CIKM Workshops*
Singapore Police Force. (2020a). *Police News Release: Annual Crime Brief 2019*. Retrieved August 5, 2020a
Singapore Police Force. (2020b). *Police News Release: Mid-Year Crime Statistics*. Retrieved August 29, 2020b, from Singapore Police Force: https://www.police.gov.sg/Media-Room/Statistics
Smith, R. G., & Jorna, P. (2011). Fraud in the 'outback': Capable guardianship in preventing financial crime in regional and remote communities. *Trends and Issues in Crime and Criminal Justice,* 413(1)
Software Freedom Conservancy. (2013). Selenium Webdriver documentation
Tompson, L., & Chainey, S. (2011). Profiling illegal waste activity: using crime scripts as a data collection and analytical strategy. *European Journal on Criminal Policy and Research, 17*, 179–201.
Vishwanath, A. (2015). Examining the distinct antecedents of e-mail habits and its influence on the outcomes of a phishing attack journal of computer-mediated. *Communication, 20*, 570–584.
Wang, J., Herath, T., Chen, R., Vishwanath, A., & Rao, H. R. (2012). Research article phishing susceptibility: an investigation into the processing of a targeted spear phishing email. *IEEE Transactions on Professional Communication, 55*(4), 345–362.
Wilsem, J., & v. (2011). Worlds tied together? Online and non-domestic routine activities and their impact on digital and traditional threat victimization. *European Journal of Criminology, 8*(2), 115–127.
Wright, R., & Marett, K. (2010). The influence of experiential and dispositional factors in phishing: an empirical investigation of the deceived. *Journal of Management Information Systems, 27*(1), 273–303.
Yee, Z., Yeh, V., Ong, S., & Han, Y. (2019). Stealing more than just your heart: a preliminary study of online love scams. *Home Team Journal—By Practioners, For Practioners* (8).

Publisher's Note