Uncovering Modus Operandi: Using unsupervised NLP algorithms to identify patterns in unstructured crime data

Group F5

David Jung (13646168), Eveline Kalff (10802657), Lizzy Da Rocha Bazilio (11426594), Lois Rink (14028417) and Mannes Mokkenstorm (11922222)
Data System Project 2021-2022
University of Amsterdam

ABSTRACT

Current crime solving techniques are failing to keep up with crime development and require further improvements to comprehend existing crime schemes. The goal of crime analysts is to establish a better understanding of crime problems to eventually be able to perform crime interventions and make crime predictions to protect social order and safety. This research explores multiple unsupervised Natural Language Processing (NLP) techniques to extract information and identify patterns in unstructured court verdicts from the Netherlands. To identify semantic structure in unstructured text, a Latent Dirichlet Allocation (LDA) analysis, Latent Semantic Analysis (LSA), Scattertext, and TF-IDF analyses are employed. These analyses enable categorization of textual data into topic clusters. The final model provides the Police Academy with a tool that extracts a structured data set of court verdicts and presents insights on topics and its relationships within different crime areas.

Link to code: https://github.com/Lizzydrb/modus-operandi-f5-py

KEYWORDS

Modus Operandi, Natural Language Processing, Unsupervised Learning, Court verdicts, LDA, LSA, TF-IDF, Scattertext

1 INTRODUCTION

Crime has always been one of the primary threats to society, and effective crime investigation is therefore a crucial task necessary to maintain a safe environment [34]. In the Netherlands, the National Dutch Police is responsible for maintaining social order and upholding the law, as well as undertaking investigation of suspected and committed crimes [1]. Data scientific efforts have been taken to establish a better understanding of criminal behaviour, and ultimately predict criminal activity [10, 13, 29]. By analysing crime data, analysts focus on discovering recurring patterns of specific behaviour [10, 34]. These behavioural patterns can indicate "a particular way or method of doing something" [19] and can be referred to as modus operandi. Identifying modus operandi proves to be a difficult task because of the unstructured nature of crime data and the lack of advanced techniques to recognize criminal behaviour from textual data [13, 18]. Current crime analysis techniques are failing to keep up with crime development and require further improvement to comprehend existing crime schemes and perform effective crime interventions [10, 14, 34]. As a sub-organization of the National Dutch Police, the Police Academy is focused on exploring methods for such improvements. Central to their goal is "training, knowledge and research for the Dutch Police" [2].

To obtain new perspectives on extracting modus operandi, the Police Academy, together with the Netherlands Organization for Applied Scientific Research (TNO) [4], has assigned data science students at the University of Amsterdam to investigate methodologies that can help in crime analysis. With the purpose of tackling the aforementioned challenges of identifying modus operandi, this research will aim to develop a tool to help structure crime data and ultimately extract crime topic clusters. The analysis incorporates Natural Language Processing (NLP) techniques on crime-related data, of which the results will be demonstrated in a prototype dashboard containing various visualizations. Because crime data of the Dutch Police is proprietary, this study scraped open-source data from rechtspraak.nl [3]. Rechtspraak-data contain real-life verdicts of the criminal courts in the Netherlands, and is thus closest to crime data from the police. Any confidential information has already been blurred on Rechtspraak, so there will be no ethical concerns. Unsupervised text-mining algorithms will be used to filter relevant information from the data and implement some structure. Subsequently, topic modelling techniques are applied to the filtered data set to identify semantic structure in the collection of textual data. Specifically, a Latent Dirichlet Allocation (LDA) approach is employed. LDA analyses enable textual data to be categorized into topics in a probabilistic manner, which allows a verdict to have multiple topics and also captures the heterogeneity of these topics [11]. The Term Frequency - Inverse Document Frequency (TF-IDF) score subsequently determines the importance of a word to a verdict in the complete crime data set [8]. Furthermore, this research applies a Latent Semantic Analysis (LSA), which is another unsupervised classification algorithm that uses computational statistics to determine the relationship between verdicts and the terms they contain. The LSA produces a set of concepts that relate to the verdicts and terms [32]. In addition, this work implements the visualisation of two classes in a corpus using a scatterplot like visualisation called scattertext. This allows to efficiently analyze, for example, two years or two types of crimes and provides insights about distinctive words within each class.

This report will first define the problem that the Police Academy currently faces and formulate research questions accordingly. The second section highlights related literature to determine the larger scope of crime analysis challenges. In the methodology, the design process and development of the final tool are described. This section includes details on data acquisition, structuring of the data and the various analyses. The final model of this research is evaluated in the results and will be presented in a dashboard that can be utilized by crime analysts to establish some structure into complex crime data, and obtain a first insight into the topics of any data set.

Problem Statement

The Police Academy and TNO cooperate with the University of Amsterdam to perform research into new ideas on how data scientific models can detect modus operandi from crime data. Improvements in the current methodology are necessary to strengthen efforts to understand criminal behaviour. The tool proposed in this report is to be used by crime analysts working at the Police Academy.

Crime analysts are currently faced with large volumes of crime data that they manually analyse to find modus operandi. Human resources are very limited in analysing such volumes of data, let alone identify significant patterns [10, 13, 18]. Manual analyses are susceptible to human mistakes and biases, where personal interpretations can influence the outcome [16]. Moreover, the distribution among various analysts can result in important information to be missed because of the variety of factual knowledge [13]. Data scientific tools have significantly more processing power, and can analyse the available crime data as a whole [34]. Furthermore, such tools can provide a complete overview of the data and provide more objectiveness [16]. Regardless, the unstructured nature of crime data remains a challenge and causes difficulties in extracting relevant information from crime data.

The Police Academy requires a tool that can help crime analysts structure and understand their data efficiently. There were no specific requirements that had to be met for the final model, besides that it must aid in understanding modus operandi. To ensure that the Police Academy can easily implement the proposed model to their own data sets, this study aims to provide a generalizable and robust tool. Moreover, it is important that the model remains flexible for possible adaptations and diverse applications. For example, the final tool must be adjustable for various crime types or topics. The proposed model will provide crime analysts with a method to gain a first insight into crime data, upon which they can later build their further research into modus operandi. Based on this, the following research question is formulated: To what extent can unsupervised NLP algorithms be used to extract patterns in unstructured crime data?

2 RELATED LITERATURE

Crime analysts focus on discovering modus operandi to predict crime [34], link crime scenes to each other, and identify criminal networks [13]. Besides the complexity of the crime data itself, crime is often not systematic, nor random [10, 18, 29]. This brings an additional challenge in the identification of criminal patterns, and can be problematic with current methodologies. Current methods namely often involve manual keyword searches across crime data [10, 13, 29]. These keywords have been predetermined, either based on previously identified modus operandi, or probability sampling referred to as dip sampling [10]. Dip sampling methods extract a random sub-set of data to analyse and determine patterns that can be identified with specific keyword searches [10]. Both of these approaches can raise concerns when identifying current crime schemes. The first approach is dependent on previous knowledge of crime analysts, which would result in novel modus operandi to be left unidentified. The second is susceptible to bias and could potentially neglect smaller clusters of modus operandi [10]. Again, certain modus operandi groupings might be left undetected [14].

Previous work has used topic modelling techniques to uncover crime topics across crime data sets. A paper by Birks et al. [10] applied LDA to perform within-crime classifications to ultimately aid the process of identifying specific criminal patterns. In the research, Birks et al. focus on creating groupings of crimes within existing crime topics that represent various modus operandi and can therefore help crime analysts in their comprehension of current crime problems. Overall, LDA has already successfully been applied in multiple research fields [30], and has proven a robust tool for topic modelling [6, 10]. Moreover, the assumption of LDA that documents contain multiple topics, and that topics consist of words that occur jointly, provides the opportunity to perform analyses on crime-related data [10].

LSA is another unsupervised classification algorithm that is often used in the field of Information Retrieval and text mining, and has proven to be effective in crime analysis [23, 35]. The intuition behind LSA is that all the contexts in which a term occurs (and also does not occur) establishes restrictions that indicate how similar the meaning of words and groups of words are [32]. In other words, terms that are semantically similar often appear in comparable documents. Through this, LSA can uncover hidden semantic relations between and among words, sentences, documents and topics in data sets [23, 35]. Subsequently, by considering similar concepts within textual data, it can capture topics in crime data [23], and also filter noise in texts [35].

Both the LDA and LSA are robust classification models, however, lack in providing a microscopic and quantitative view of the documents. For crime analysts it is important to be able to investigate how crime develops in order to identify potential changes in crime schemes [34]. For this type of investigation, a quantitative analysis is useful to statistically compare crime development over the years. TF-IDF is a vector space model that uses statistical methods to analyze text similarity. The goal of this measure is to quantify the importance of terms, with the intuition that terms that occur in less documents are more informative and vice versa. By measuring the term frequency and inverse document frequency, we can analyze how rare a word is in the whole data set. With this, TF-IDF can provide an overview of the importance of each term in a document relative to the whole text corpora [8].

3 METHODOLOGY

Data

To design a tool that uses NLP algorithms to detect patterns in unstructured crime data, we first extracted court verdicts that were made publicly available through www.rechtspraak.nl. Rechtspraak.nl is the official government website, containing a combination of information about legal proceedings and anonymous transcripts of court rulings. To ensure a final data set that contains enough data, verdicts from multiple years were collected, starting with the most recent available year (2021). Following Birks et al. (2020) [10], we aimed for a data set of approximately 9.000 instances, which could be obtained by taking all criminal law (in Dutch: *strafrecht*) verdicts from these years. These were downloaded as raw XML-files and pre-processed to create a data set with data and metadata of the verdicts.

Manual inspection of the raw XML-files showed that the data made available through rechtspraak.nl was structured differently for each courthouse. This is shown in the irregular presence and absence of section titles, numbering of sections, XML structure and the reference to (often absent) appendices. While rechtspraak.nl contains many verdicts, some of those were empty or contained very little textual data. To filter these out, a choice was made to include only those verdicts which contain more than 1000 characters.

Since our research goal is to find Modus Operandi in unstructured text, we only took criminal law verdicts into account. Other law areas where for example civil rights, administrative law and public law. Moreover, a large proportion of the verdict, contained information about the trial itself and about the consequences of the (possible) conviction. This information does not provide insight in the Modus Operandi of criminals and should therefore be excluded from the analysis. The sections in the verdicts that did contain information about the Modus Operandi where the indication of content, evidence and indictment (in Dutch: *inhoudsindicatie, bewijs* and *tenlastelegging*). The analysis that was conducted for this research is based on the text in these three sections.

If a verdict complies with the previously described requirements, the raw-XML data is parsed to extract the verdict date, crime date, ECLI reference and the courthouse. The content indication of verdict and the evidence and indictment as presented during the trial are extracted using RegEx [7]. All verdicts for which either an indictment or evidence could be extracted were included in the final data set that consists of 19.976 verdicts. This is more than the 9.000 as described in Birks et al. (2020) [10] where their research is based on only one crime type. Since our goal is to provide the stakeholder with a robust tool that can help them to gain insight in the Modus Operandi in unstructured text about various crime areas, we decided to collect more data, from multiple crime types.

Data preprocessing

Before one can use textual data for different NLP algorithms, texts should be properly preprocessed. The programming language Python [33] was used for all the coding in this work, and the following preprocessing methods were applied:

- **Tokenization** with the Python package *Gensim* [27]
- **Stop words removal** with the python package *nltk* [9]. The list can be manually adjusted
- Creation of **bigrams** with the Python package *Gensim*
- Creation of **trigrams** with the Python package *Gensim*
- **Lemmatization** with the Python package *SpaCy* [17]

Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) is an unsupervised classification algorithm and an example of topic modelling introduced by Andrew (2001) [11] which discovers topics in a collection of documents. More specifically, it is a generative probabilistic model for collections of discrete data, such as text corpora. Topics are considered to be a collection of terms which taken together propose a latent topic.

Notion and Terminology. Although the LDA model is not tied to textual data, it is most commonly used for it, as it is in this work. Therefore, it can be helpful for certain readers to clarify frequently used terms such as 'word', 'document' and 'corpus'.

- **Word:** A word is a basic unit of discrete data which is defined to be an item from a larger vocabulary indexed by $\{1, ..., V\}$. Unit-basis vectors are being used for representing a single word where all components equal 0 except one single component which equals to 1.
- **Document:** A document is a sequence of N words which is being denoted by $w = (w_1, w_2, ..., w_M)$ where w_n is the nth word in the sequence.
- **Corpus:** A collection of M words is referred to as corpus, which is denoted by $D = \{w_1, w_2, ..., w_M\}$.

Dirichlet Distributions. Some readers might not be familiar with Dirichlet distributions. Therefore, a brief description within the context of LDA is being given in this section.
Suppose three topics are present in a theoretical corpus. Therefore, LDA creates a triangle where each corner represents a certain topic and positions all documents within the area of the triangle close to the corner a document belongs to. If all three topics are equally apparent within a document, it would be positioned in the middle of the triangle. The question is how the LDA algorithm positions the documents within that triangle. This question will be answered in the following sections.

A Dirichlet distribution is being defined by one parameter, which is in this paper α & β. If $\alpha = 1$, the Dirichlet distributions looks as in Figure 1. Therefore, by drawing random samples from such a distribution, one can observe that those samples are evenly distributed across the surface area.

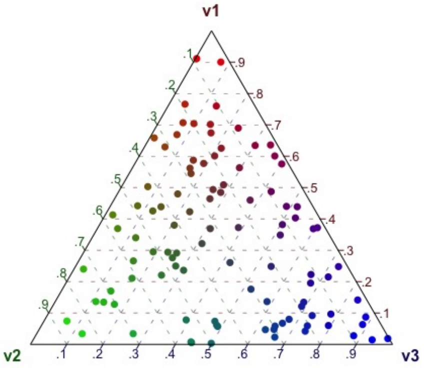


Figure 1: Visualization of Dirichlet distribution with $\alpha = 1$

Choosing an $\alpha > 1$, in this case 10, one can observe that drawn samples from this distribution lead to an aggregation in the centre. This can be seen in Figure 2.

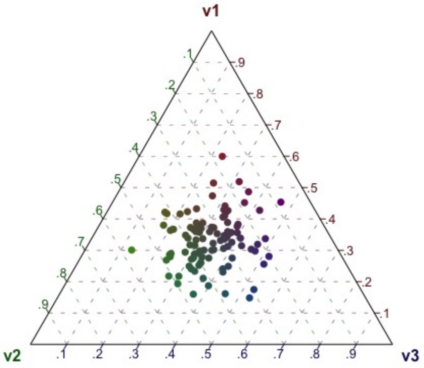


Figure 2: Visualization of Dirichlet distribution with $\alpha = 10$

Lastly, choosing an $\alpha < 1$, 0.1 in this case, the probability of drawing random samples close to the corners is higher which can be seen in Figure 3.

The LDA model. The goal of the LDA model is to find a generative probabilistic model of a corpus, which not only assigns a high probability to a member of the corpus, but also assigns high probability to other similar documents. The basic idea behind the model is that documents are represented as a random mixture of latent topics. Each topic is characterized by a distribution over words [11]. The following generative process for each document in a corpus is assumed by LDA:

(1) Choose $N \sim \text{Poisson}(\xi)$
(2) Choose $\theta \sim \text{Dir}(\alpha)$, normally $\alpha < 1$, Choose $\varphi \sim \text{Dir}(\beta)$

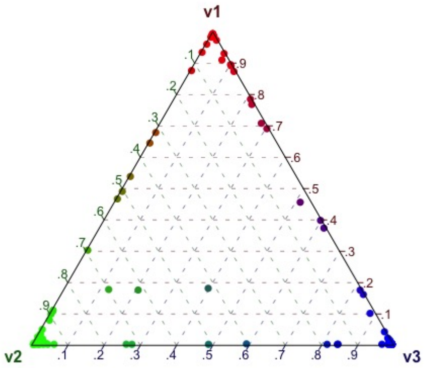


Figure 3: Visualization of Dirichlet distribution with $\alpha = 0.1$

(3) For each of the N words w_n:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$
   (b) Choose a word $w \sim \text{Multinomial}(\varphi)$

A plate notation of the LDA model can be seen in Figure 4 and the mathematical formula describing the probability of a document in Equation 1. Please note that rectangles, or so-called plates, in a plate notation visualisation represent replicates which are repeated entities. The outer rectangle represents documents, whereas the inner rectangle the repeated word positions in a given document [21].

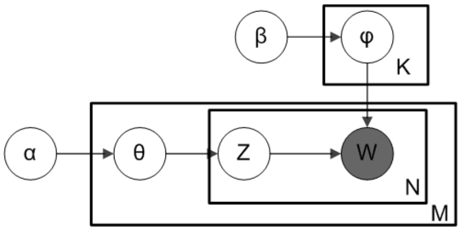


Figure 4: Graphical model representation of the LDA model

$$P(W, Z, \theta, \varphi, \alpha, \beta) = \prod_{j=1}^{M} P(\theta_j; \alpha) \prod_{i=1}^{K} P(\varphi_i; \beta) \prod_{t=1}^{N} P(Z_{j,t}|\theta_j) P(W_{j,t}|\varphi z_{j,t}) \tag{1}$$

The name of the variables are defined as follows:

- M: Number of documents
- N: Number of words in a given document
- α: Parameter of the Dirichlet distribution (topic)
- β: Parameter of the Dirichlet distribution (word)
- θ: Topic distribution for documents
- φ: Word distribution
- z: Topic of a word
- w: Word
- K: Number of topics

Equation 1 basically consists of four factors, where the first, as well as the second factor, are Dirichlet distributions and the third and fourth are Multinomial distributions. The first and third factors describe topics, and the second and fourth words. For further technical details, please take a look at Andrew (2001) [11]. However, a simplified version of how the LDA algorithm operates is being explained in the following section.

Procedure of the LDA algorithm. Point (1) of the generative process introduced in the previous section stands for the length of the document. Thus, the length of a document is drawn from a Poisson distribution with parameter ξ. Once the length of the document has been determined, one can move to point (2) which is being represented by the first factor in Equation 1 which is a Dirichlet distribution such as the one in Figure 3. A random sample from such a distribution is being drawn, which leads to the topic for this document. A sample output could look like this: (0.7, 0.1, 0.2). Each number stands for a percentage of the corresponding topic. These percentages are used to create a Multinomial distribution in (3.a) which is the third factor in Equation 1. With the help of this Multinomial distribution, topics can be drawn for every word within the document. Since the number of words has already been determined by the Poisson distribution, one can draw N samples from this Distribution with replacement.

Once the topic of each word has been drawn, it is time to draw words corresponding to those topics. This is being done with the second Dirichlet distribution, which is the second factor in Equation 1 which is one that associates the topics to words. This can be visualized in the same manner as in Figure 1 - 3, but now the corners are no longer topics but words. If only four words were apparent in a corpus, a sample output of a certain topic could look as follows: (0.4, 0.4, 0.1, 0.1). One can see that this topic is positioned in the middle between word 1 and 2 and further away from word 3 and 4. This Dirichlet distribution is being used to create another Multinomial distribution like before, which is the last factor in Equation 1. These percentages represent the probability of a certain word occurring within that topic. This Multinomial distribution is used for drawing words from the already drawn topics from the first Multinomial distribution, and therefore one can finish the document.

Term frequency-Inverse Document Frequency

The term frequency-Inverse Document Frequency (TF-IDF) is considered and empirical method and is one of the most commonly used term weighting schemes in modern information retrieval systems [8]. The algorithm consists of two parts where the first part is the term frequency (TF) which comes down to the number of times each term occurs in each document. Note that 'term' equals the definition of 'word' as described in the previous paragraph. A formula for the term frequency is given below, where the term is annotated with t, document with d and frequency with f:

$$tf_{t,d} = \frac{f_{t,d}}{\sum_{t' \in d} f'_{t,d}} \tag{2}$$

The second part is the inverse document frequency (IDF) which is a measure about the amount of information a word contains, if it is common or rare across all documents. The IDF is calculated by taking the logarithmically scaled inverse of the fraction of the documents (D) that contain a certain word:

$$idf_{t,D} = log\frac{N}{df_d} \tag{3}$$

The equation that links the TF with the IDF is given in the formula below.

$$tfidf_{t,d} = \frac{f_{t,d}}{\sum_{t' \in d} f'_{t,d}} * log\frac{N}{df_j} \tag{4}$$

Latent Semantic Analysis

Latent semantic analysis (LSA) is, just like LDA, an unsupervised classification algorithm which extracts and represents the contextual-usage meaning of terms by applying computational statistics to a large corpus of text [32]. The algorithm uses two steps covering the decomposition of a matrix of documents and terms using Singular value decomposition (SVD). First, a document-term matrix is generated in formula with the use of the TF-IDF method described in equation 4 where a weight is assigned to a word t in document d.

The next step holds the matrix factorization with SVD, where the document-term matrix is decomposed into three parts to extract all different dimensions. The formula below shows the matrix factorization, where m denotes the documents, n the unique words in the documents and r the number of topics. The different dimensions are the document-topic matrix (U), the word-topic matrix (V) and the diagonal matrix of the singular values ($\sum$). The SVD method applied on the TF-IDF matrix as formulated below reveals the underlying semantic relationship between terms and documents [22].

$$A_{mxn} = U_{mxr} * \sum_{rxr} * V^T_{rxn} \tag{5}$$

LSA is great for automatic categorization of a large corpus of text and for search engine optimization, with its main aspects as addressed by Hoenkamp (2011) [15]:

(1) Recovering latent semantic factors underlying the document space.
(2) Compression with loss of the document space by eliminating lexical noise.
(3) By applying SVD, the lexical noise is eliminated.

The algorithm is, however, limited to the ability to capture multiple meanings of a word and the use of an unordered collection of words in the form of a bag of words model (BOW). The larger the document space, thus the BOW, the less likely LSA will find an optimal set of semantic topics [22].

4 SCATTERTEXT

Scattertext is an open-source tool for visualizing linguistic variation between document categories in a language-independent way and was developed by [20]. The backbone of the tool is a scatterplot where the axes correspond to the rank-frequency a term occurs in a category of documents. Please note that within the context of scattertext, bigrams, trigrams etc. are being referred to as terms. Scattertext is built on tinytext [36] and inspired by Ruder (2014) [28]. A set of unigrams and bigrams that are found in the corpus of documents assigned to one of the two categories are plotted on a two-dimensional scatter plot.

A corpus of documents is defined as C with disjoint subsets A and B s.t. $A \cup B \equiv C$. Let $\phi^T(t, C)$ be the number of times term t occurs in C, $\phi^T(t, A)$ be the number of times t occurs in A. Let $\phi^D(t, A)$ refer to the number of documents in A containing t. Let $t_{i,j}$ be the jth word in term t_i. Normally, $j \in \{1, 2\}$. The parameter ϕ may be ϕ^T or ϕ^D. One can also use different feature representations for ϕ such as TF-IDF [20].

$$Pr[t_i] = \frac{\phi(t_i, C)}{\sum_{t \in C \wedge |t| \equiv |t_i|} \phi(t, C)} \tag{6}$$

To construct the set of terms which are included in visualisation V, one has to follow a two-step process. For a term to be included, it must occur at least m times. Let:

$$PMI(t_i) = log\frac{Pr[t_i]}{\prod_{t_{i,j} \in t_i Pr[T_{i,j}]}} \tag{7}$$

be the PMI (pointwise mutual information). The minimum PMI accepted is p. Thus, V can be defined as:

$$\{t|\phi(t, C) \geq m \wedge (|t| \equiv 1 \vee PMI(t) > p)\} \tag{8}$$

(x^A_t, x^B_t) are term t's coordinates on the scatterplot where A and B are the two document categories. Although x^K_t is proportional to $\phi(t, K)$, a large number of terms will have identical $\phi(t, K)$ values. Breaking ties is being done by defining larger x^K_t values to words that appear alphabetically last. Let us define t^K_t s.t. $t \in V$ and $K \in \{A, B\}$ as the ranks of $\phi(t, K)$ which are sorted in ascending order, where ties are broken by terms' alphabetical order [20]. This leads to:

$$x^K_t = \frac{r^K_t}{argmax\ r^K} \tag{9}$$

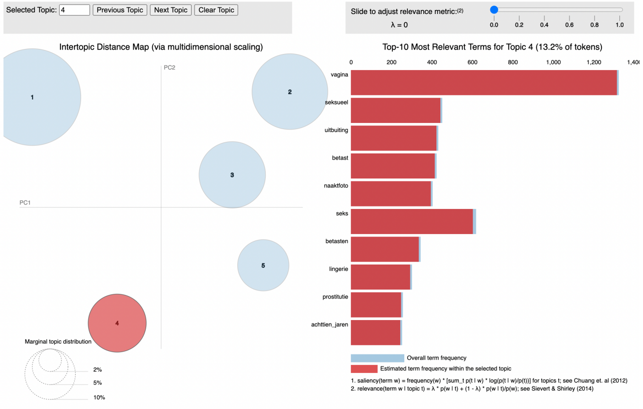
Thereby, x values are limited to [0,1] and both axes are scaled identically.

5 RESULTS

After successful cleaning of the data and the applied algorithms mentioned in the previous section, the following results were obtained.

LDA

With the help of the python package *pyLDAvis* [31] one can obtain the following visualisation of the created LDA model.
Topics are displayed on the left-hand side and can be analysed in greater detail by clicking on the topic or searching for it in the search bar in the top-left corner. By selecting a topic, the ten most relevant terms within that topic are being displayed on the right-hand side in descending order from top to bottom. One can also use different lambda λ values for filtering out words which are frequently to be found across all topics and therefore of less interest. Values of lambda close to zero filter out those words and values close to one leave them in the visualisation. Please note that five topics were searched for in Figure 5 and are represented by numbers from 1-5. The user itself must determine the topics according to the

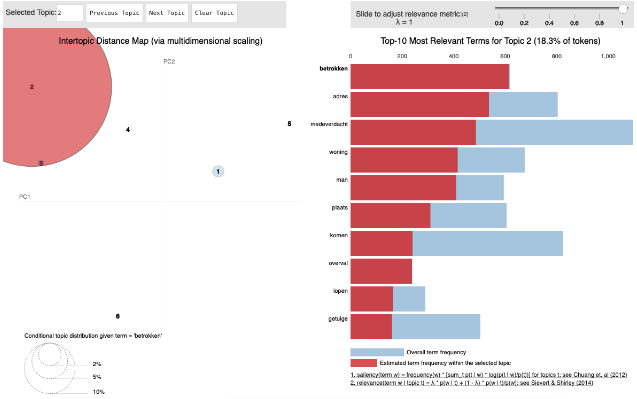


1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 5: Visualisation of LDA model

terms presented to the right.

By selecting a word on the right-hand side, the size of the circles on the left-hand side are being adjusted according to the frequency of how the selected term is also of importance in the other topics. This can be seen in Figure 6 where the term *betrokken* is selected. One can see that this term is basically only apparent in one topic.


Selected Topic: 2
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
2
3
4
5
1
6
Conditional topic distribution given term = 'betrokken'
2%
5%
10%
Top-10 Most Relevant Terms for Topic 2 (18.3% of tokens)
0
200
400
600
800
1,000
betrokken
adres
medeverdacht
woning
man
plaats
komen
overval
lopen
getuige
Overall term frequency
Estimated term frequency within the selected topic

1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 6: Visualisation of selected term

LSA

Once the LSA algorithm is performed on the data, an output with ten terms with corresponding TF-IDF scores that represent topics is generated. The overall performance of the algorithm is better on smaller data sets. This confirms the limitation of the algorithm mentioned in the literature. However, the algorithm is useful for visualisations of topics even for bigger data sets. To visually display the output of the found topics in the data, a data frame was conducted with two columns. The first column contains all the words that belong to a topic, where each topic is represented by ten words. The second column contains the topic number each word belongs to. With the help of *visdcc* in *Dash* a network graph which requires lists of nodes and edges as an input can be created. The 'word' column is set as an input list for the nodes and the 'topic' column is set as an input list for the edges. In figure 7 an example is given for the crimetype *harddrugs*. Four subnetworks are found within the data and each network is connected to another network. This visualisation greatly displays the subtopics found and the similarities between them. Just as with LDA, the user must name the subtopic found according to the terms that occur in the network graphs.

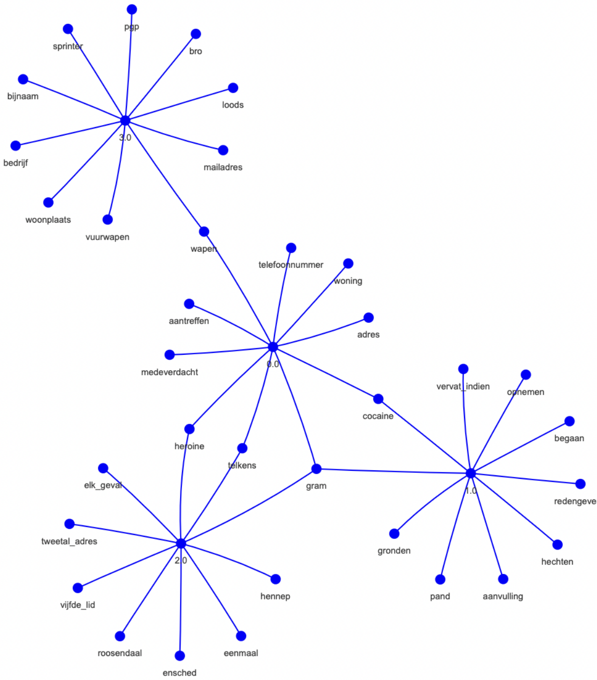


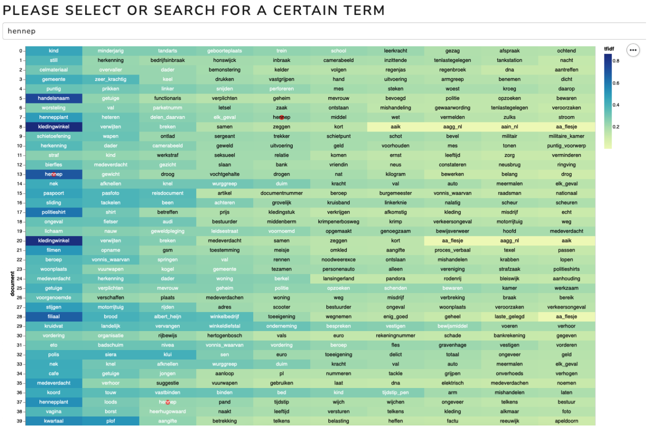
Figure 7: Visualisation of LSA model on crime type rape

LDA overall performs better than LSA based on the coherence score, but in some cases LSA with improvement of TF-IDF has a higher coherence score and thus both algorithms should be considered. It can therefore be concluded that the use of the algorithms is case-specific and so is the performance.

TF-IDF Heatmap

LDA and LSA detect topics amongst documents, which provides a macroscopic interpretation about the corpus. To obtain a microscopic interpretation, one would like to be able to gain a quick overview about each document. This can be done with a heatmap, where each row represents a document and each column a term within that document. The terms are being arranged according to their TF-IDF score in descending order from left to right. Thus, the term with the highest TF-IDF score can be found in the first column.

This visualisation allows the user to get a quick intuition about each document. If one is looking for a specific term, one can use the dropdown menu or search bar in the top left corner and search for the desired term which will be highlighted with a red dot which can be seen in Figure 8. This allows one to quickly find the term.



PLEASE SELECT OR SEARCH FOR A CERTAIN TERM

hennep

document										
0	kind	minderjarig	tandarts	geboorteplaats	trein	school	leerkracht	gezag	afspraak	ochtend
1	still	herkenning	bedrijfsinbraak	honswijck	inbraak	camerabeeld	inzittende	tenlastegelegen	tankstation	nacht
2	celmateriaal	overvaller	dader	bemonstering	kelder	volgen	regenjas	regenbroek	dna	aantreffen
3	gemeente	zeer_krachtig	keel	drukken	vastgrijpen	hand	uitvoering	armgreep	benemen	dicht
4	puntig	prikken	linker	snijden	perforeren	mes	steken	woest	kroeg	daarop
5	handelsnaam	getuige	functionaris	verplichten	geheim	mevrouw	bevoegd	politie	opzoeken	bewaren
6	worsteling	val	parketnumm	letsel	zaak	ontstaan	mishandeling	gewaarwording	tenlastegelegen	veroorzaken
7	hennepplant	heteren	delen_daarvan	elk_geval	hennep	middel	wet	vermelden	zulks	stroom
8	kledingwinkel	verwijten	breken	samen	zeggen	kort	aaik	aagg_nl	aain_nl	aa_flesje
9	schietoefening	wapen	ontlad	sergeant	trekker	schietpunt	schot	bevel	militair	militaire_kamer
10	herkenning	dader	camerabeeld	geweld	uitvoering	geld	voorhouden	mes	tonen	puntig_voorwerp
11	straf	kind	werkstraf	seksueel	relatie	komen	ernst	leeftijd	zorg	verminderen
12	bierfles	medeverdacht	gezicht	slaan	bank	vriendin	neus	constateren	neusbrug	ringving
13	hennep	gewicht	droog	vochtgehalte	drogen	nat	kilogram	bewerken	belang	drog
14	nek	afknellen	knel	wurggreep	duim	kracht	val	auto	meermalen	elk_geval
15	paspoort	pasfoto	reisdocument	artikel	documentnummer	beroep	burgemeester	vonnis_waarvan	raadsman	nationaal
16	sliding	tackelen	been	achteren	grovelijk	kruisband	linkerknie	nalatig	scheur	scheuren
17	politieshirt	shirt	betreffen	prijs	kledingstuk	verkrijgen	afkomstig	kleding	misdrijf	echt
18	ongeval	fietser	audi	bestuurder	middenberm	krimpenerbosweg	krimp	verkeersongeval	motorrijtuig	weg
19	lichaam	nauw	geweldpleging	leidsestraat	voornoemd	opgemaakt	genoegzaam	bewijsverweer	hoofd	medeverdacht
20	kledingwinkel	verwijten	breken	medeverdacht	samen	zeggen	kort	aa_flesje	aagg_nl	aaik
21	filmen	opname	gsm	toestemming	meisje	omkled	aangifte	proces_verbaal	texel	passen
22	beroep	vonnis_waarvan	springen	val	rennen	noodweerexce	ontslaan	mishandelen	krabben	lopen
23	woonplaats	vuurwapen	kogel	gemeente	tezamen	personenauto	alleen	vereniging	strafzaak	politieshirts
24	medeverdacht	herkenning	dader	woning	berkel	lansingerland	pandora	rodenrij	bleiswijk	aanhouding
25	getuige	verplichten	mevrouw	geheim	politie	opzoeken	schenden	bewaren	kamer	werkzaam
26	voorgenoemde	verschaffen	plaats	medeverdachen	woning	weg	misdrijf	verbreking	braak	bereik
27	stijgen	motorrijtuig	rijden	adres	scooter	bestuurder	ongeval	woonplaats	veroorzaken	verkeersongeval
28	filiaal	brood	albert_heijn	winkelbedrijf	toeeigening	wegnemen	enig_goed	geheel	laste_gelegd	aa_flesje
29	kruidvat	landelijk	vervangen	winkeldiefstal	onderneming	bespreken	vestigen	bewijsmiddel	voeren	verhoor
30	vordering	organisatie	rijbewijs	hertogenbosch	vals	euro	rekeningnummer	schade	bankrekening	gegeven
31	eto	badschuim	nivea	vonnis_waarvan	vordering	beroep	fles	gravenhage	vestigen	vorderen
32	polis	siera	klui	sen	euro	toeeigening	delict	totaal	ongeveer	geld
33	nek	knel	afknellen	wurggreep	duim	kracht	val	auto	meermalen	elk_geval
34	cafe	getuige	jongen	aanloop	pl	nummeren	tackle	grijpen	onverhoeds	verhogen
35	medeverdacht	verhoor	suggestie	vuurwapen	gebruiken	laat	dna	elektrisch	medeverdachen	noemen
36	koord	touw	vastbinden	binden	bed	kind	tijdstip_pen	arm	mishandelen	laten
37	hennepplant	loods	hennep	pand	tijdstip	wijch	wijchen	ongeveer	telkens	bestuur
38	vagina	borst	heerhugowaard	naakt	leeftijd	versturen	telkens	kleding	alkmaar	foto
39	kwartaal	plof	aangifte	betrekking	telkens	belasting	heffen	factu	reeuwijk	apeldoorn

Figure 8: Heatmap of TF-IDF scores of words per document

TF-IDF Score over time

For further analysis, one can also investigate how the TF-IDF score of a specific word has changed over time, which can be seen in Figure 9.

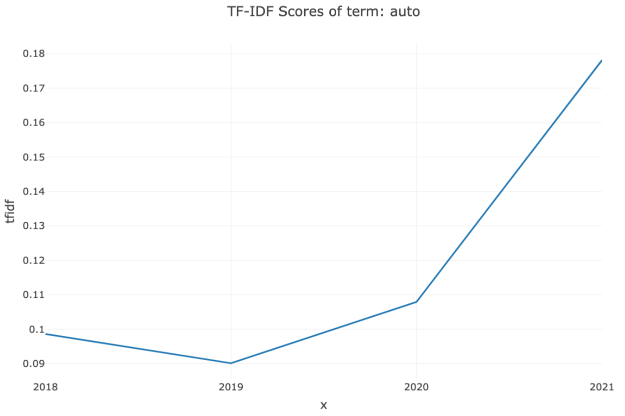


Figure 9: Line plot of TF-IDF score of selected word

Scattertext

Scattertext efficiently visualizes two classes and their corpora in a scatterplot like visualisation according to the term frequencies within each class. Figure 10 shows the comparison between all the words of the years 2018 and 2019 where the Y-axis is the term frequency of 2018 and the X-axis is the term frequency of 2019 and range from 'Infrequent' to 'Frequent'. Each dot represents a word within the documents from 2018 or 2019. The closer a dot is to the top of the plot, the more frequently it occurred in 2018. The further right a dot, the more that word occurred in 2019. In order so preserve computational power, a threshold has been implemented in the scattertext algorithm to filter out terms that occur highly infrequently. However, if a term was never present in neither of the selected years, it would be placed in the bottom-left corner. Words that occur frequently in both years appear in the top-right corner.

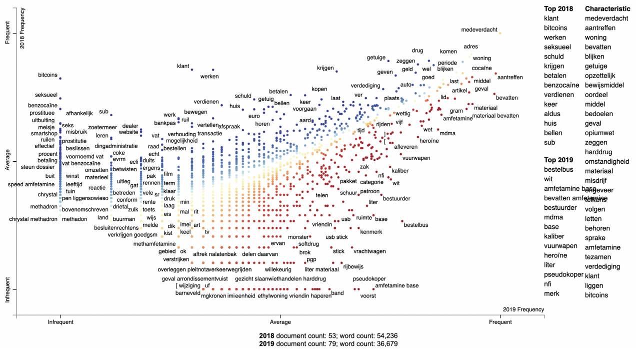
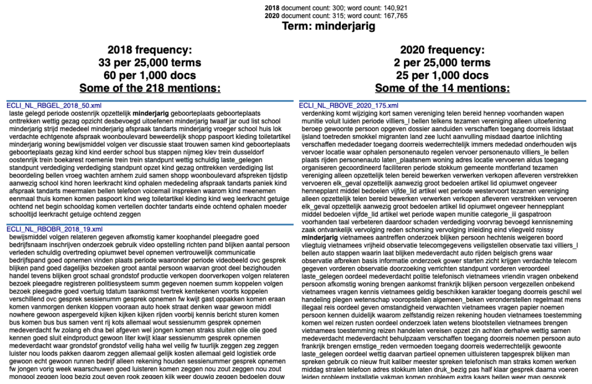


Figure 10: Scattertext visualisation of 2018 and 2019

Of great interest are the upper-left and lower-right corners, where words are being placed that uniquely represent the corresponding year. A good example can be seen in Figure 10 where in the top-left corner the word *bitcoin* appears. *Bitcoin* therefore occurred frequently in 2018 but infrequently in 2019. This makes sense since the peak of the bullrun of bitcoin occurred at the beginning of 2018 [12]. Typically, around the peaks many frauds happen, which is being detected by scattertext in this case.

One can also select or search a certain word which will be used for scanning all documents. All the documents which contain the selected word are being listed down below with the selected word highlighted. In Figure 11 one can see an example of a selection of documents which contain the word *minderjarig* (minor). This allows the user to make a quick comparison between documents and classes.



2018 document count: 300; word count: 140,921
2020 document count: 315; word count: 167,765

Term: minderjarig

2018 frequency:
33 per 25,000 terms
60 per 1,000 docs
Some of the 218 mentions:

ECLI_NL_RBGEL_2018_50.xml

laste gelegd periode oostenrijk opzettelijk **minderjarig** geboorteplaats geboorteplaats onttrekken wettig gezag opzicht desbevoegd uitoefenen minderjarig twaalf jar oud list school minderjarig strijd mededeel minderjarig afspraak tandarts minderjarig vroeger school huis lok verdachte echtgenote afspraak woonboulevard beweerdelijk shopp paspoort kleding toiletartikel minderjarig woning bewijsmiddel volgen ver discussie staat trouwen samen kind geboorteplaats geboorteplaats gezag kind kind eerder school bus stappen nijmeg klev trein dusseldorf oostenrijk trein boekarest roemenie trein trein standpunt wettig schuldig laste_gelegen standpunt verdediging verdediging standpunt opzet kind gezag onttrekken verdediging list beoordeling bellen vroeg wachten arnhem zuid samen shopp woonboulevard afspreken tijdstip aanwezig school kind horen leerkracht kind ophalen mededeling afspraak tandarts paniek kind afspraak tandarts meermalen bellen telefoon voicemail inspreken waarom kind meenemen eenmaal thuis komen komen paspoort kind weg toiletartikel kleding kind weg leerkracht getuige ochtend net begin schooldag komen vertellen dochter tandarts einde ochtend ophalen moeder schooltijd leerkracht getuige ochtend zeggen

ECLI_NL_RBOBR_2018_19.xml

bewijsmiddel volgen relateren gegeven afkomstig kamer koophandel pleegadre goed bedrijfsnaam inschrijven onderzoek gebruik video opstelling richten pand blijken aantal persoon verleden schuldig overtreding opiumwet bevel opnemen vertrouwelijk communicatie bedrijfspand goed opnemen vinden plaats periode waaronder periode videobeeld ovc gesprek blijken pand goed dagelijks bezoeken groot aantal persoon waarvan groot deel bezighouden handel tevens blijken groot schaal grondstof productie verkopen doorverkopen volgen relateren bezoek pleegadre registreren politiesysteem summ gegeven noemen summ koppelen volgen bezoek pleegadre goed voertuig tdatum taankomst tvertrek kentekenen voorts koppelen verschillend ovc gesprek sessienumm gesprek opnemen fw kwijt gast oppakken komen eraan komen vanmorgen denken kloppen vooraan auto hoek straat denken waar gewoon middl nowhere gewoon aspergeveld kijken kijken kijken rijden voorbij kennis bericht sturen komen bus komen bus bus samen vent rij kots allemaal wout sessienumm gesprek opnemen medeverdacht fw zolang eh dna bel afgeven wel jongen komen straks sluiten olie olie goed kennen goed sluit eindproduct gewoon liter kwijt klaar sessienumm gesprek opnemen medeverdacht waar grondstof grondstof veilig haha wel veilig fw tuurlijk zeggen zeg zeggen luister nou loods pakken daarom zeggen allemaal gelijk kosten allemaal geld logistiek orde gewoon echt gewoon runnen bedrijf alleen rekening houden sessienummer gesprek opnemen fw jongen vorig week waarschuwen goed luisteren komen zeggen nou zout zeggen nou zout mongool zeggen loog bezig zout geven rook zeggen kijk weer douwig zeggen bedoelen douw

2020 frequency:
2 per 25,000 terms
25 per 1,000 docs
Some of the 14 mentions:

ECLI_NL_RBOVE_2020_175.xml

verdenking komt wijziging kort samen vereniging telen bereid hennep voorhanden wapen munitie voluit luiden periode villiers_l bellen telkens tezamen vereniging alleen uitoefening beroep gewoonte persoon opgeven dossier aanduiden verschaffen toegang doorreis lidstaat ijsland toetreden smokkel migranten land zee lucht aanvulling misdaad daartoe inlichting verschaffen mededader toegang doorreis wederrechtelijk immers mededad onderhouden wijs vervoer locatie waar ophalen personenauto regelen vervoer personenauto villiers_le bellen plaats rijden personenauto laten_plaatsnem woning adres locatie vervoeren aldus toegang organiseren gecoordineerd faciliteren periode stokkum gemeente montferland tezamen vereniging alleen opzettelijk telen bereid bewerken verwerken verkopen afleveren verstrekken vervoeren elk_geval opzettelijk aanwezig groot bedoelen artikel lid opiumwet ongeveer hennepplant middel bedoelen vijfde_lid artikel wet periode westervoort tezamen vereniging alleen opzettelijk telen bereid bewerken verwerken verkopen afleveren verstrekken vervoeren elk_geval opzettelijk aanwezig groot bedoelen artikel lid opiumwet ongeveer hennepplant middel bedoelen vijfde_lid artikel wet periode wapen munitie categorie_iii gaspatroon voorhanden taal verbeteren daardoor schaden verdediging voorvrag bevoegd kennisneming zaak ontvankelijk vervolging reden schorsing vervolging inleiding eind vliegveld roissy **minderjarig** vietnamees aantreffen onderzoek blijken persoon hechtenis weigeren boord vliegtuig vietnamees vrijheid observatie telecomgegevens veiligstellen observatie taxi villiers_l bellen auto stappen waarin laat blijken medeverdacht auto rijden belgisch grens waar observatie afbreken basis informatie onderzoek gower starten zicht krijgen verdachte telecom gegeven vorderen observatie doorzoeking verrichten standpunt vorderen veroordeel laste_gelegen oordeel medeverdacht politie telefonisch vietnamees vriendin vragen onbekend persoon afkomstig woning brengen aankomst frankrijk blijken persoon vergezellen onbekend vietnamees vragen kennis vietnamees geldig beschikken karakter toegang doorreis geschil wel handeling plegen wetenschap vooropstellen algemeen_beken veronderstellen regelmaat mens illegaal reis oordeel geven omstandigheid verwachten vietnamees vragen papier noemen persoon kennen duidelijk waarom zelfstandig reizen rekening houden vietnamees toestemming komen wel reizen rusten oordeel onderzoek laten wetens blootstellen vietnamees brengen vietnamees toestemming reizen handelen vereisen opzet zin achten derhalve wettig samen medeverdacht medeverdacht behulpzaam verschaffen toegang doorreis noemen persoon auto frankrijk brengen ernstige_reden vermoeden toegang doorreis wederrechtelijk gewoonte laste_gelegen oordeel wettig daarvan partieel opnemen uitluisteren tapgesprek blijken man spreken gebruik co nieuw fruit kaliber meester spreken telefonisch man straks komen werken middag stralen telefoon adres stokkum laten druk_bezig pas half klaar gesprek daarna voeren leiden probleem installatie vakman komen probleem extra kaars bellen weer man gesprek

Figure 11: Documents containing 'minderjarig'

6 VALIDATION OF APPROACH

Since unsupervised NLP algorithms are used on unstructured data, one is limited in the selection of validation methods. To validate the obtained results from LDA and LSA for topic modelling, this work used the 'Coherence Score'. The coherence score measured the degree of interpretability of the created topics to humans. This measurement helps to distinguish between topics that are semantically interpretable topics and topics which are artefacts of statistical inference. It is important to note that the coherence score is highly dependent on the data and therefore cannot be used for comparison between models which use different data. However, it can be used to compare results that use the same data. There are currently two popular choices of coherence scores.

CV Coherence Score

The CV coherence score creates content vectors (CV) of words using their co-occurrences. Once the vectors are created, the coherence score is calculated using normalized pointwise mutual information (NPMI) and the cosine similarity. By default, the *Gensim* package uses this version of the coherence score. The CV score can take the following values: $0 \leq \text{c\_v} \leq 1$ [5, 24]. However, CV behaves poorly when it is being used for randomly generated word sets. A more robust version is the UMass Coherence Score, which will be described in the following section.

UMass Coherence Score

This coherence score is calculated how often two words w_i and w_j appear together in the corpus and was introduced by [25] and is defined as follows.

$$C_{UMass}(w_i, w_j) = log \frac{D(w_i, w_j) + 1}{D(w_i)} \tag{10}$$

- $D(w_i, w_j)$: Number of times words w_i and w_j appear together in a document
- $D(w_i)$: Number of times word w_i appeared alone in a document
- Important: $C_{UMass}(w_i, w_j) \neq C_{UMass}(w_j, w_i)$

The UMass score can take the following values: $-14 \leq \text{u\_mass} \leq 14$. Both c_v and u_mass scores should be maximized. The results can be seen in Table 1.

Perplexity Score

One can also use the perplexity score for an LDA model, which captures how surprised a model is of new data which it has not seen before and is measure as the normalized log-likelihood of a held-out test set. For LDA, a test set is a collection of unseen documents w_d and the model described by the topic matrix Φ and the hyperparameter α for the Dirichlet distribution of documents [26]. Therefore, one has to evaluate the following log-likelihood:

$$\mathcal{L}(w) = log\, p(w|\Phi, \alpha) = \sum_d logp(w_d|\Phi, \alpha) \tag{11}$$

This likelihood uses a set of unseen documents w_d given the topics Φ and the hyperparameter α for the Dirichlet distribution θ_d of documents. This measure can be taken for the held-out documents w_d and is defined as:

$$Perplexity(test\ set\ w) = exp(\frac{\mathcal{L}(w)}{count\ of\ words}) \tag{12}$$

The lower the perplexity, the better the model [26]. For the other algorithms used in this work, there are no validation metrics.

The results of the classification algorithms (Table 1) show that based on the coherence scores, the two models perform equally on our data set. However, no firm conclusion based on the output can be drawn since the algorithms performance is case-specific. Therefore, both algorithms should be considered.

7 DISCUSSION

By means of NLP techniques, this study generated a novel tool to extract crime topic clusters from unstructured verdict data from Rechtspraak, with the objective to provide more insight into crime topics. Crime is a threat to everyone in society, as well as a society itself. Therefore, every citizen benefits directly or indirectly from the work that the police, and in turn Police Academy, does on a daily basis. Any improvements to the techniques used for crime analysis therefore provides usefulness to society.

Based on the initial research question, the report has indicated that the proposed tool can indeed be used to extract patterns in unstructured crime data. LDA provides a global overview of the topics in the data, TF-IDF indicated a more microscopic and fundamental measurement of the terms and its importance in documents, scattertext provides an overview of classes in the data, and LSA enables the capturing of semantic relations between words and documents. Because this study specifically focused on creating a generalizable and flexible model, the final tool can easily be implemented in practice by crime analysts at the Police Academy. The tool was designed in such a way that it can be applied to different data sets, for example confidential data from the Police, and its flexibility allows crime analysts to alter the analyses for various crime areas or topics. In fact, the tool itself is not restricted to crime data and can thus be used for any textual data. For example, the Public Health Service of Amsterdam (GGD) or the fire brigade could utilize this tool for their data and gain valuable insights. However, the tool also has some limitations.

The data from Rechtspraak that was used in the current study, introduces some bias as a consequence of its structure and missing data. Because of the lack of coherent structure of the data, it was not possible to extract relevant information from all the verdicts in the data set. In an attempt to capture deviations, the extraction-approach was modified to the most common document structure. Consequently, the data set that was ultimately used for the analyses is somewhat incomplete. Documents that contain relevant information, but have a different structure, were not included in the current data set, which could potentially lead to bias towards certain crime- or court areas that follow the specific structure that was defined in this report. Future work should focus on creating more regular expressions to allow the capturing of a higher number

Table 1: Coherence scores and perplexity for LDA and LSA

data set	LDA			LSA	
	coherence score c_v	coherence score u_{mass}	perplexity	coherence score c_v	coherence score u_{mass}
Total data	0.480	-0.961	-7.574	0.471	-1.776
2018	0.432	-0.825	-7.310	0.390	-1.277
2019	0.327	-1.291	-7.496	0.423	-1.455
2020	0.453	-0.735	-7.479	0.483	-1.155
2021	0.488	-0.940	-7.516	0.506	-2.038
Burglary	0.480	-0.751	-7.067	0.499	-1.053
Harddrugs	0.471	-0.549	-7.190	0.387	-0.844
Abuse	0.590	-0.728	-7.297	0.439	-1.143
Rape	0.538	-0.805	-7.350	0.443	-1.399

of different document structures, leading to less bias in the data collection phase. In addition, rechtspraak.nl contains many incomplete verdicts that include little information. To counter for this missing data, a data set of complete verdicts that contain information about evidence and charges should be created. Overall, a well-balanced data set could decrease the biases that are currently present.

The bias in the data set can allow for misinterpretations of the reality crime schemes, and consequently give an inaccurate representation of modus operandi. It is important to acknowledge and recognize these biases for ethical concerns. The models in this report are not fit for predictive policing, and should only be used to get a better grasp and understand of current trends in court cases, which can subsequently be used for further investigations.

Future research should also investigate the possibility of using Named Entity Recognition (NER) as an additional investigation tool, as this would allow the mapping of words and their contextual use. To successfully implement this, a large data set with court verdicts should be annotated by human annotators to measure the performance of the automatic NER on court verdicts specifically since they follow an entirely different structure than other texts.

8 CONCLUSION

This study's aim was to provide crime analysts at the Police Academy with an approach to extract relevant information about modus operandi from unstructured crime data, and present an intuitive tool that can give insights into crime data by capturing crime topics. The data contained verdicts from rechtspraak.nl that were filtered by identifying some structure in the documents, and subsequently structured using unsupervised text-mining techniques.

The tool introduced in this work is capable of delivering first insights into crime data, regarding modus operandi, in an intuitive manner. The tool is generalizable so that it can be applied to different data sets, as well as flexible to ensure analyses within multiple crime areas. In addition to that, since generalizability lies in its nature, it can easily be extended for different fields of interest in the future.

REFERENCES

[1] [n.d.]. *Politie.* https://www.politie.nl/en

[2] [n.d.]. *Politie Academy.* https://www.politieacademie.nl/english

[3] [n.d.]. *Rechtspraak.* https://www.rechtspraak.nl

[4] [n.d.]. *TNO.* https://www.tno.nl/en/

[5] [n.d.]. Topic coherence pipeline. https://radimrehurek.com/gensim/models/coherencemodel.html. Accessed: 2022-02-07.

[6] 2018. Machine learning algorithms and police decision-making legal. http://www.excellenceinpolicing.org.uk/wp-content/uploads/2018/09/1-4_NewTech_Law_Privacy_Ethics.pdf

[7] Alfred V Aho. 1991. Algorithms for finding patterns in strings, Handbook of theoretical computer science (vol. A): algorithms and complexity.

[8] A. Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing Management* 39(1) (2003), 45–65.

[9] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[10] Daniel Birks, Alex Coleman, and David Jackson. 2020. Unsupervised identification of crime problems from police free-text data. *Crime Science* 9, 1 (2020), 1–19.

[11] David Blei, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning Research* 3 (2003), 993–1022.

[12] Juan Cárdenas-Rodríguez, Mateo Restrepo Sierra, and David Plazas Escudero. 2018. Cryptocurrency Scams.

[13] Vikas Grover, Richard Adderley, and Max Bramer. 2007. Review of Current Crime Prediction Techniques. *Applications and Innovations in Intelligent Systems XIV* c (2007), 233–237. https://doi.org/10.1007/978-1-84628-666-7_19

[14] Malith Munasinghe Harsha Perera, Shanika Udeshini. 2014. Criminal Short Listing and Crime Forecasting Based on Modus Criminal Short Listing and Crime. (2014).

[15] E. Hoenkamp. 2011. Trading Spaces: On the Lore and Limitations of Latent Semantic Analysis. *Advances in Information Retrieval Theory* (2011), 40–51.

[16] Monica C Holmes and Diane D Comstock-davidson. 2007. Data Mining and Expert Systems in Law Enforcement Agencies. *Issues In Information Systems* VIII, 2 (2007), 329–335. https://doi.org/10.48009/2_iis_2007_329-335

[17] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. (2017). To appear.

[18] Wolfgang Jentner, Dominik Sacha, Florian Stoffel, Geoffrey Ellis, Leishi Zhang, and Daniel A. Keim. 2018. Making machine intelligence less scary for criminal analysts: reflections on designing a visual comparative case analysis tool. *Visual Computer* 34, 9 (2018), 1225–1241. https://doi.org/10.1007/s00371-018-1483-0

[19] 2020. *The Oxford English Dictionary.* Oxford University Press, Oxford, United Kingdom.

[20] Jason S. Kessler. 2017. Scattertext: a Browser-Based Tool for Visualizing how Corpora Differ. (2017).

[21] D. Koller and N. Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press. https://books.google.co.in/books?id=7dzpHCHzNQ4C

[22] Hevia A. Weber R. Rios S. L'Huillier, G. 2010. Latent semantic analysis and keyword extraction for phishing classification. *IEEE International Conference on Intelligence and Security Informatics* (2010).

[23] Kaiz Merchant and Yash Pande. 2018. NLP Based Latent Semantic Analysis for Legal Text Summarization. *2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018* (2018), 1803–1807. https://doi.org/10.1109/ICACCI.2018.8554831

[24] Sara Mifrah and EL Habib Benlahmar. 2020. Topic Modeling Coherence: A Comparative Study between LDA and NMF Models using COVID'19 Corpus. *International Journal of Advanced Trends in Computer Science and Engineering* (08 2020). https://doi.org/10.30534/ijatcse/2020/231942020

[25] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing Semantic Coherence in Topic Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (Edinburgh, United Kingdom) *(EMNLP '11)*. Association for Computational Linguistics, USA, 262–272.

[26] Quentin Pleplé. [n.d.]. Perplexity To Evaluate Topic Models. http://qpleple.com/perplexity-to-evaluate-topic-models/. Accessed: 2022-02-07.

[27] Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic* 3, 2 (2011).

[28] Christian Rudder. 2014. *Dataclysm: Who We Are (When We Think No One's Looking)*. Crown Publishing Group, USA.

[29] Shiju Sathyadevan, M. S. Devan, and S. Surya Gangadharan. 2014. Crime analysis and prediction using data mining. *1st International Conference on Networks and Soft Computing, ICNSC 2014 - Proceedings* (2014), 406–412. https://doi.org/10.1109/CNSC.2014.6906719

[30] Yao Xie Shixiang Zhu. 2018. Crime Incidents Embedding Using Restricted Boltzmann Machines. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* (2018), 2376–2380. https://doi.org/10.1109/ICASSP.2018.8461621

[31] Carson Sievert and Kenneth Shirley. 2014. LDAvis: A method for visualizing and interpreting topics. In *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. Association for Computational Linguistics, Baltimore, Maryland, USA, 63–70. https://doi.org/10.3115/v1/W14-3110

[32] Peter W. Foltz Thomas K Landauer and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes* 25:2-3 (1998), 259–284.

[33] Guido Van Rossum and Fred L Drake Jr. 1995. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

[34] Nisal Waduge. 2017. Machine Learning Approaches For Detect Crime Patterns. September (2017), 1–7.

[35] Canyu Wang, Xuebi Guo, and Hao Han. 2012. Crime detection using Latent Semantic Analysis and hierarchical structure. *ICSESS 2012 - Proceedings of 2012 IEEE 3rd International Conference on Software Engineering and Service Science* (2012), 337–340. https://doi.org/10.1109/ICSESS.2012.6269474

[36] Yihui Xie. 2019. TinyTeX: A lightweight, cross-platform, and easy-to-maintain LaTeX distribution based on TeX Live. *TUGboat* 1 (2019), 30–32. https://tug.org/TUGboat/Contents/contents40-1.html

List of Figures

List of Tables