Malicious Interlocutor Detection Using Forensic Analysis of Historic Data

Michael Seedall

A thesis submitted to the University of Huddersfield in partial fulfilment of the requirements for the Degree of Master of Science By Research

The University of Huddersfield

January 2022

Acknowledgements

I would like to thank Dr Katrinna Macfarlane for her on-going support, guidance, and initial inspiration to undertake the research. Although the journey through the research has proved challenging at times given the subject matter, our on-going aspirations of realising a safer internet for children has spurred me on.

I would also like to thank Dr Violeta Holmes for her guidance, encouragement, wisdom, and support throughout the research.

I must acknowledge Ian McManus for his tireless endeavour in helping me to collate, process, and parse terabytes of chat corpora data at Blackburn University Centre which formed part of this research.

Thank you to Preverted-Justice.com for providing me with open access to their datacentre. Without the data from their datacentre the research would not have been possible.

And finally, family, whose support throughout the journey has been pivotal to pushing on and getting through.

Copyright Statement:

I. The author of this thesis (including any appendices and/ or schedules to this thesis) owns any copyright in it (the "Copyright") and s/he has given The University of Huddersfield the right to use such Copyright for any administrative, promotional, educational and/or teaching purposes.

II. Copies of this thesis, either in full or in extracts, may be made only in accordance with the regulations of the University Library. Details of these regulations may be obtained from the Librarian. Details of these regulations may be obtained from the Librarian. This page must form part of any such copies made.

III. The ownership of any patents, designs, trademarks and any and all other intellectual property rights except for the Copyright (the "Intellectual Property Rights") and any reproductions of copyright works, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property Rights and Reproductions cannot and must not be made available for use without permission of the owner(s) of the relevant Intellectual Property Rights and/or Reproductions.

Abstract

The on-going problem of child grooming online grows year on year and whilst government legislation looks to combat the issue by levying heavier penalties on perpetrators of online grooming, crime figures still increase. Government guidance directed towards digital platforms and social media providers places emphasis on child safety online. As this research shows, government initiatives have proved somewhat ineffective. Therefore, the aim of this research is to investigate the scale of the of the problem and test a variety of machine learning and deep learning techniques that could be used in a novel intelligent solution to protect children from online predation.

The heterogeneity of online platforms means that a one size fits all solution presents a complex problem that needs to be solved. The maturity of intelligent approaches to Natural Language Processing makes it possible to analyse and process text data in a wide variety of ways. Pre-processing data enables the preparation of text data in a format that machines can understand and reason about without the need for human interaction.

The on-going development of Machine Learning and Deep Learning architectures enables the construction of intelligent solutions that can classify text data in ways never imagined. This thesis presents research that tests the application of potential intelligent solutions such as Artificial Neural Networks and Machine Learning algorithms applied in Natural Language Processing. The research also tests the performance of pre-processing workflows and the impact of pre-processing of both online grooming and more general chat corpora. The storage and processing of data via a traditional relational database management system has also been tested for suitability when looking to detect grooming conversation in historical data.

The research has tested the performance of Artificial Neural Networks, Recurrent Neural Networks, and Convolutional Neural Networks in sentiment analysis tasks to establish the overall sentiment of online grooming conversation. The results of the tests conducted have proved successful and have provided opportunity for future work adapting neural networks to classification tasks related to detecting online grooming conversation.

Document similarity measures such as Cosine Similarity and Support Vector Machines have displayed positive results in identifying grooming conversation, however, a more intelligent solution may prove to have better currency in developing a smart autonomous solution given the ever-evolving lexicon used by participants in online chat conversations.

Publications

2019:

Seedall, M., Macfarlane, K. & Holmes, V., 2019. *SafeChat System with Natural Language Processing and Deep Neural Networks.* Huddersfield, EMiT/University of Huddersfield/High End Compute Ltd/University of Manchester , pp. 28-31.

2021:

Macfarlane, K., Seedall, M & Holmes, V., 2021. An Ensemble Learning Approach to Autonomous Agent Decision Making, Paper Pending.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 Introduction

The World Wide Web commonly known as "the internet" has become a diverse, dynamic, ever evolving and interactive medium that has allowed humans to interact and communicate in ways never imagined. From the inception of the internet in 1969 as a simple means of communication over the TCP/IP protocols, no one could have imagined the rapid growth and dynamic pace of development in Web technologies and platforms. (Brittanica, 2020)

The internet has presented new opportunities in eCommerce, collaboration, cloud, information services, education, science, industry, and beyond. Often termed as the "4th Industrial Revolution" the internet has transformed the ways in which humans conduct business, learn, interact, and communicate in a social context (Xu, et al., 2018). Whilst the internet could be deemed as a motivator for the greater good of humanity it may not always facilitate or harbour the ethos of "for the good of mankind". One of the fastest growing mediums to arise from this dynamic and diverse World Wide Web is that of social interaction facilitated by social media platforms. Modern platforms often contain online chat mediums which are an evolution of the early online chat environments such as bulletin boards and chat rooms. One harrowing activity to evolve from such mediums is the activity of "online sexual grooming of children" and therefore forms the focus of this research.

The rapid and continued growth of mobile technologies and end devices in general, has led to the proliferation of ubiquitous computing where, worryingly, the interaction with such ubiquitous resources such as mobile phones, tablet devices and other end devices has seen children under the age of five using a connected computing device. In some cases, children under the age of 5 have a social media online presence. (Ofcom, 2020)

The regulatory body that regulates the communications industry - *Office of Communications* (Ofcom) discussed in their 2020 report "Children's Media Lives – Wave 6", that there are substantial numbers of children under the age of 15 that have a social media online presence even if this means that they are under the minimum age of 13 required for the social media platform. *Ofcom* also state that "by the age of 15 almost all children have a social media account". Whilst children have access to such social media platforms and other chat environments there is a real present danger of children being exposed to risk of the actions of a malicious actor online (Ofcom, 2020). However, under the *Convention on the Rights of the Child* (UNCRC), *United Nations Children's Fund* (UNICEF) state in article 17 of the convention that "every child has the right to reliable information from a variety of sources, and governments should encourage the media to provide information that children can understand. Governments must help protect children from materials that could harm them" (UNICEF, 2009). One means in which children can access mass media resources is using the internet. Article 17 places responsibility on governments to ensure there is appropriate legislation in place that keeps children safe from harmful content.

There is a moral, ethical, and legal requirement for parents, government(s) and other agencies to protect children online to keep them safe whilst allowing them to fulfil their human right to have access to the internet and the wealth of information it contains. The changing landscape of social media platforms, changes in technology and proliferation of technologies available to children presents a range of challenges in employing effective protection and detection strategies.

1.1 Background Problem

The text-based Computer Mediated Communication (CMC) paradigm is not a recent phenomenon and therefore has a place in the annals of computing history. The first implementations of CMC were used within the US military, government, and universities with the transmission of the first email between two networked computers taking place in 1972. (Hafner & Lyon, 1996)

The popularity of CMC grew exponentially in the late 1980's to early 1990s when Internet Service Providers (ISP) allowed the public to get online. This led to what Herring (2010) states was "the golden age" of CMC. This golden age saw the growth in multiparticipant text interactions through electronic mailing lists, Usenet newsgroups, Multiuser Dimension/Dungeon/Dialogues (MUDs), Multi-user-Dungeon Object Orientated (MOOS), and Internet Relay Chat (IRC). With the evolution of handheld mobile devices using the Short Messaging Service (SMS) CMC once again enjoyed a period of growth. More recently, the emergence of multimedia, social media and other content sharing platforms has ensured the continued engagement with CMC. (Herring, 2010; Ingram et al., 2000)

Herring, (2010) posits that reference to CMC is now such that users often refer to interactions as conversations just as they would in the physical sense. Further to this, Herring points out that as early as 1987 several scholars had begun to refer to CMC explicitly as "conversation". This acceptance of CMC as conversation could suggest a normal way of doing things and such communication is no different from that held in the physical sense. There would seem to be no differentiation between the digital and physical discourse in terms of how users now view their interactions. The exponential expansion of the web and modern ubiquitous computing paradigms have facilitated "everybody being connected to everyone else" through high-speed networks, mobile networks, chat rooms, social media, text, and video interactions for example. The evolution of the web has also seen an upsurge in cybercrime and for the purpose of this research an upsurge in online grooming of children.

On the 13th June 2016 the British Broadcasting Company (BBC) reported that a 15-year-old school girl in Leicestershire was raped and killed by a man who was twice her age. The initial contact was via the popular social media platform – Facebook. The article goes on further to explain that even though there was initial contact via Facebook messenger service the pair had swapped mobile numbers within 10 minutes and then went on to exchange 2600 messages mainly by text message. (BBC, 2016)

On 5th July 2019 BBC Wales reported on a case where a man from Pontypridd had been jailed for grooming 146 children online. The perpetrator of this prolific offence had used nine different *Facebook* accounts as well as three gaming accounts using a range of strategies including luring some of his victims with the promise of providing them with gaming credits. He used software to create the persona of children posing as young boys or girls. He went on to incite children as young as eight years old in sexual activity online. The groomer used fake profiles on a range of other platforms to give him access to children. The activities were often recorded and then shared those recordings with other paedophiles online (BBC, 2019).

A further case reported on by the *Daily Star* newspaper ran with the following headline:

> "Dad-of-two tried to meet girl, 8, for sex equipped with pack of condoms and tent."

In this attempt to meet with the child the groomer had been using the social media application Kik to coerce what he thought was an 8-year-old girl. Luckily, in this instance the groomer had been engaged in discourse with undercover police officers who arranged to meet the offender and subsequently arrested him. The groomer was found with condoms, a tent, alcohol, and cannabis and thus was meeting with the full intention of meeting the underage girl and engaging in sexual activity. (Star, 2020)

The *National Society for the Prevention of Cruelty to Children* (NSPCC) reported in 2019 that statistics it had obtained under the freedom of information act revealed that up to March 2019:

- 5,161 crimes related to sexual communication with children had been reported within England and Wales in the previous 18 months.
- There had been a 50% increase in offences recorded in the previous 6 months.
- The was a 200% rise in instances of Instagram being used to target children during the same period.

(NSPCC, 2019)

Under their campaign "Wild West Web" the NSPCC look to lobby the UK government to address the issue of online grooming with the introduction of an Online Harms Bill which aims to make technology companies accountable for the online abuse facilitated by their platforms. The campaign is lobbying for the following:

- New rules to make tech companies put safety first.
- Punishments for failing to protect young users.
- Safer social platforms that tackle online abuse.

(NSPCC, 2020)

The NSPCC go on to discuss that overall, children are spending more time online and are potentially receiving less interaction and supervision from what it claims are "over stretched" online moderators.

Children have been spending more time online with potentially less supervision, and less intervention from over stretched online moderators. This has exposed children to an increased risk of online abuse. Data compiled by *Europol* shows significant increases in activity relating to child sexual abuse and exploitation, including a rise in the number of referrals from the *National Centre for Missing and Exploited Children* to *Europol* about child sexual abuse material. The *Internet Watch Foundation* (IWF) received an increase of 50% in reports of online child sexual abuse. (Bentley, et al., 2020)

The IWF reported that it had 44,809 reports of online abuse between 23rd March 2020 and 9th July 2020 compared to 29,698 reports of abuse during the same period in 2019. (IWF, 2020). The IWF also reports that 8.8 million users in the UK attempted to view child sexual abuse online and goes on to discuss that the findings of *The National Crime Agency* (NCA) are that there are currently 300,000 individuals in the UK that pose a threat to children in the categories of Physical Contact or Online Abuse. (IWF, 2020)

This rise in reporting and apparent rise in could be due in part and have a direct correlation to the increased uptake and use of mobile and other connected end devices by children.

Ofcom's "Children's Media Lives – Wave 6" report 2020 found that in relation to social media platforms in 2019:

- Many of the children sampled used YouTube and Snapchat daily.
- Within the sample older aged girls predominantly used Snapchat.
- The application known as TikTok used for lip synching videos proved very popular with younger children.

The study also found that *WhatsApp* had grown in popularity in the age group 12–15-year-old even though the minimum age for *WhatsApp* is 16 Years.

(Ofcom, 2020)

In a previous report in 2016 *Ofcom* found that one in five 8 to 11-year-old children and seven in ten 12 to 15-year-old have a social media presence. The study also found that in the target age group of 5- to 15-year-olds 48% of children had use of or owned a smartphone device. Ofcom (2016)

An analysis of mobile ownership by age was carried out by *Statista.com* in 2020 which found the following results displayed in Figure 1.

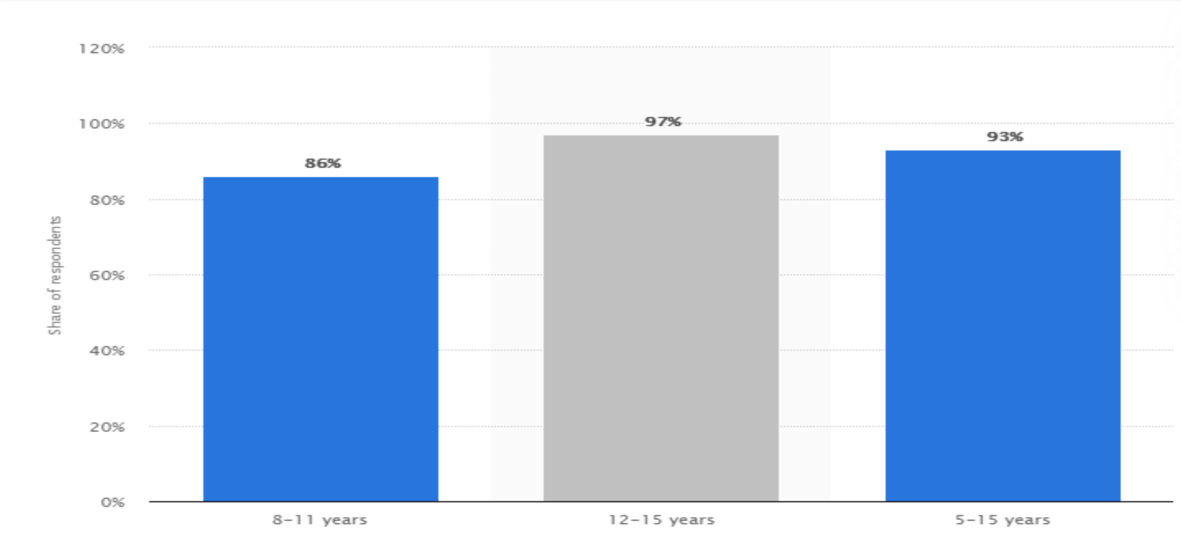


Figure 1: Share of children owning tablets and smartphones in the United Kingdom (UK) 2019, by age. Reproduced from (Statista.com, 2020)

The worrying statistic here is the apparent proliferation of mobile device ownership in such vulnerable age groups.

1.2 Motivation for this research.

This research exists to investigate ways in which children and other vulnerable groups can exist safely in a digital world without fear from malicious actors whose sole purpose is to cause harm. However, to grasp the scale of the issue of online grooming there needs to be discussion relating to the size of the problem through the identification of the current level of online grooming crimes as well as how such crimes are carried out.

News articles across a range of mediums portray a grim landscape in the types of stories reported with no real solution to the early detection and the protection of young people online. Technologies grow apace and the interaction with these technologies and the platforms children interact with displays the same upward trend.

There must be consideration of the state of current measures implemented by government and other stakeholders as well as the characteristics of the vulnerable individuals who are victims of such crime. Furthermore, there must be an understanding of the modus operandi of the perpetrators of such crime, which in turn must influence and drive the design of intelligent and autonomous detection systems.

Children have the right to access online material not only for leisure but also for education and development of wider social skills beyond those of physical interactions. Children also have a right to privacy online and must have a degree of autonomy and the ability to express themselves in safe environment. In their industry guidance titled “Children’s Online Privacy and freedom of Expression”, Unicef, (2018) outline five guiding principles for a shared responsibility of a variety of actors from governments and businesses to parents and educators. Those five principles are as follows:

1. Children have the right to privacy and the protection of their personal data.
2. Children have the right to freedom of expression and access to information from a diversity of sources.
3. Children have the right not to be subjected to attacks on their reputation.
4. Children's privacy and freedom of expression should be protected and respected in accordance with their evolving capacities.
5. Children have the right to access remedies for violations and abuses of their rights to privacy and free expression, and attacks on their reputation.

(UNICEF, 2018)

On-going developments in Artificial Intelligence and long-established approaches in both machine learning and deep learning in the field of Natural Language Processing make it possible to develop mature intelligent systems that can perform a wealth of different operations on digital data. The apparent lack of detection systems being developed and deployed by the large social media platforms, even though one would think there is a corporate responsibility to protect all sections of a digital society, has provided a great source of motivation for this research and its intended aims and objectives.

1.3 Aims and Objectives

The aim of this research is to investigate the current state of UK legislation, child use and presence on the web, online social platforms, natural language processing, machine learning, and the deep learning architectures used in the analysis and processing of digital discourse. The research also aims to evaluate the effectiveness of a range of approaches that could be used in the future construction of a novel system to detect online predation and protect children / vulnerable young adults online.

It could prove to be a mammoth undertaking to develop an autonomous plugin that would act as a sentinel for all online applications given the heterogeneous landscape and the wide range of chat platforms and applications. Therefore, the focus of this research is centred around the data generated from online chat conversations and whether machine learning and deep learning techniques could provide positive identification of chat interactions as they take place on the web. Ideally, this identification/classification of grooming discourse would happen in real-time, and this research aims to test whether such real-time detection could be a possibility in a future detection system. The following research question is therefore proposed:

"Malicious Interlocutor Detection Using Forensic Analysis of Historic Data."

In order to address the research question the following objectives were identified:

- Review the state of current research in the fields of Artificial Intelligence, Machine Learning, Deep Learning and Natural Language Processing (NLP).
- Collect, pre-process and process chat corpora data (general and predator) for evaluation and analysis.
- Critically evaluate the effectiveness of pre-processing techniques used in NLP.
- Critically evaluate the performance of selected machine learning and deep learning approaches used in sentiment analysis and similarity detection of predatory Computer Mediated Communications (CMC).

- Critically evaluate the suitability of Relational Database Management Systems (RDBMS) as part of a chat platform and detection system.
- Based on the findings of the research, propose possible technological solution(s) to the issue of online predation in chat conversations.

1.4 Research Methods

The collection of primary research results from the analysis of data collected from the performance metrics of experiments undertaken with the goal of answering the research question proposed in section 5.4. Secondary research from a wealth of existing materials will aid in evaluating the current scale of the problem of online predation, effectiveness of current legislation, the social media landscape, and the current state of the art in NLP and the technologies that can be used in detecting online predation.

Creswell & Plano Clark (2011)defined three types of primary research as: quantitative, qualitative, and mixed methods. Various typologies or taxonomies of mixed methods had already been posited by Creswell & Plano Clark in 2011.

Johnson et al. (2007) defined mixed methods research as:

"*Mixed methods research is the type of research in which a researcher or team of researchers combines elements of qualitative and quantitative research approaches (e. g., use of qualitative and quantitative viewpoints, data collection, analysis, inference techniques) for the broad purposes of breadth and depth of understanding and corroboration."*

In this research, the types of data collected will be statistical, observational, and opinion-based methods that will also include some level of empirical research. This approach to the research therefore takes a mixed method approach. Statistical data is in the form of statistical analysis of chat corpora, comparison of results, and performance metrics of the algorithms used in machine and deep learning.

In the collection of a variety of chat corpora there will be observation of trends or quirks within the data that may not initially be identified by the various algorithms and queries run against the data. This observation and evaluation of data may, therefore, inform the design of algorithmic approaches to the processing and analysis of the various chat corpora.

The research will conduct research of the current literature in the fields of online grooming, data storage, NLP, ML, DNN, sentiment analysis, and text similarity. Comparative studies will be used to inform the research, analysis, and the design of the experiments undertaken on the online grooming data gathered.

Given the nature of online grooming data and the context in which it is used, ethical concerns surrounding the viewing of such data, especially in conducting an observational analysis of the raw text must be raised. To limit the effects on the mental health of the researcher, regular check-ins with research supervisors have been undertaken. Other ethical concerns relating to the participants in the grooming data have been considered and therefore during the lifetime of the research, participants are referred to as Predator (the adult perpetrating an online grooming crime) and Victim (the agent provocateur posing as a child online) and therefore preserve anonymity within the research undertaken.

1.5 Risk

According to Mustaro & Rossi (2013), “risk can be characterized as an event or uncertainty (due to one or more causes) that, if happens, resulting in positive or negative impact in the project”. Raftery, (1996) posits that risk can be quantified using the following measure:

Risk = Probability of event x Magnitude of loss/gain

Baccarini & Melville (2011) furthered the discussion of risk with a more thorough overview of the risks academic research may face when considering both the research participant and researcher.

Raftery’s measure and an adoption in part of Mustaro & Rossi’s approach to planning of responses to risks identified, the risks identified in this research will adopt the following approach:

- Description of risk
- Probability – low, medium, high
- Research Impact – negative or positive
- Risk mitigation strategy

(Baccarini & Melville, 2011)

The identification of risk to the research can also be aided by the use of SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis. The SWOT framework was first developed by Alfred Humphrey in the 1960s and is still a prevalent approach used today in a wide variety of situations from business, research, technology, and beyond.

Risks can, as previously mentioned, have positive and negative outcomes. Acceptance or aversion to risk can have an impact on the success or utility within a project. It could be the case that the management of risk in projects may depend on the sensitivity, impact, and nature of the project. Figure 2 below highlights the differing level of payoff/gains to be had from the differing approaches to risk acceptance in time constrained projects.

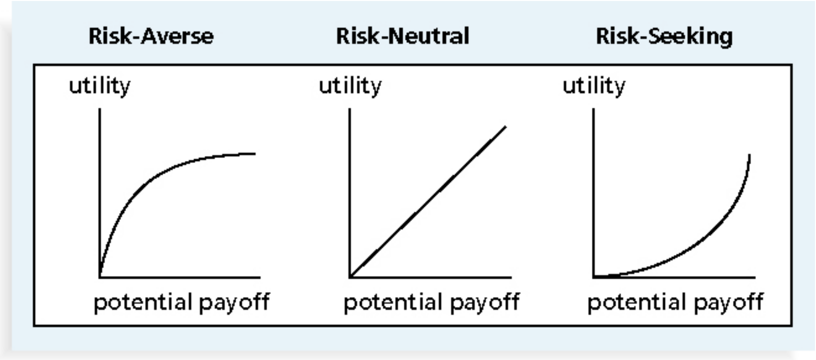


Figure 2: Risk and Utility

The research undertaken has taken the approach of risk-averse given the nature of the data utilised within the analysis of NLP, time constraints, anonymity of chat participants within the data, and the management and security of the computational resource required to undertake analysis and perform the required experiments.

1.5.1 SWOT analysis

The SWOT analysis carried out in Table 1 has informed the planning of the research carried out, the management of risk, and identify opportunities that could have a positive impact on the research.

Table 1: SWOT Analysis

SWOT ANALYSIS	
Strengths	• Access to laboratories at UCBC which are equipped with the high powered hardware resources. • Good background knowledge in managing data in an industrial setting. • Well established management of Relational Database Management Systems in both industry and educational setting. • Wealth of experience processing, storing, and managing large datasets. • Wide availability of chat corpora for analysis.
Weaknesses	• Knowledge of Python programming language in the field of AI (Machine Learning and Deep Learning). Having to learn from the ground up. • Time constraints due to external pressures (personal and professional).
Opportunities	• Exploration of new technologies. • Development of knowledge in the field of Artificial Intelligence and Natural Language Processing. • Devise solutions to identify trends and patterns in online grooming conversation. • Raise awareness to the threat of online grooming to a wider audience beyond academia. • Widen the conversation with other academics in the field of Artificial Intelligence and Natural Language Processing.
Threats	• Lack of online grooming data available for processing and analysis. • Lack of access to online grooming chat corpora. • Failure or lack of access to hardware resources at UCBC. • Prolonged illness of researcher. • Global impact of natural disaster or other catastrophic event.

1.5.2 Risk analysis

Table 2 below displays the project risk analysis carried out which is based in part on the project SWOT analysis and other considered risks to the project. Risks have been categorised based upon the Mustaro & Rossi model previously discussed.

Table 2: Risk Analysis

Risk Description	Impact Description	Impact Level Rate: 1 (low) to 5 (high)	Probaility Level Rate: 1 (low) to 5 (high)	Mitigation Strategy
Availability of online grooming data	The sourcing of true online grooming chat data fails and alternative data may not be suitable thus skewed results in the research.	3	3	Source at least 3 different corpora that are centred around different types of conversation or topic ie: Technology, Travel, general chat. Use one of the topic specific corpora as a substitute for grooming corpora and classify/detect based on the topic attributed to the substitute corpora.
Hardware Failure	Lack of access to data, Python programs, analysis and reports	5	3	Ensure alternative hardware is available and that a robust back up strategy is in place. Refer to the UCBC Computing department's incident response and disaster recovery policy. Ensure that research desktop PCs and Laptops are backed up at regular intervals.
Loss of data due to hardware failure	Failure of RDBMS or host hardware requires the rebuilding of databases or alternative data stores.	5	2	Ensure alternative hardware is available and that a robust back up strategy is in place. Refer to the UCBC Computing department's incident response and disaster recovery policy. Build high availability database cluster (Galera) to replicate data across number of servers.
RDBMS unable to process high number	Processing of data is slow and impacts on the	4	3	Move analysis to text driven data files as alternative. This may give opportunity for

of records available with chat corpora	ability to process, analyse or classify high volumes of data thus impacting on project deadlines.			comparison and critical evaluation on the two approaches within the research.
Prolonged Illness	Inability to conduct the research within the given deadlines.	5	2	Communicate with research supervisor(s) at earliest opportunity. Access relevant mechanisms available via the institution related to interruption, extension or suspension.
Catastrophic global event or natural disaster	All bets are off, the project potentially fails or is severely hampered by the event.	5	1	Communicate with research supervisor(s) at earliest opportunity. Access relevant mechanisms available via the institution related to interruption, extension or suspension.

2 Related Work

2.1 Scale of the problem and impact on UK legislation

As the cases of online sexual abuse and the on-going uptake and use of end devices by children to access a variety of web platforms there has been a need to update current UK legislation to help combat against this growing trend.

The Sexual Offences Act 2003, Section 15 sets out an offence in the case of sexual activity if a person aged 18 or over (A) commits an offence if:

"**A** has met or communicated with another person (**B**) on one or more occasions and subsequently:

(i) **A** intentionally meets **B**,

(ii) **A** travels with the intention of meeting **B** in any part of the world or arranges to meet **B** in any part of the world, or

(iii) **B** travels with the intention of meeting **A** in any part of the world,

A intends to do anything to or in respect of **B**, during or after the meeting mentioned in paragraph (i) to (iii) and in any part of the world, which if done will involve the commission by **A** of a relevant offence,

B is under 16, and

A does not reasonably believe that **B** is 16 or over. "

(Gov.uk, 2003)

In response to the growing trend of online child sexual exploitation the UK government introduced new legislation which brought into force section 67 of the Serious Crime Act 2015 in April of 2017. The legislation states "It is now a criminal offence for anyone aged 18 or over to intentionally communicate with a child under 16, where the person acts for a sexual purpose and the communication is sexual or intended to elicit a sexual response". The offence applies to online and offline communication, including social media, e-mail, texts, letters. (Gov.uk, 2017)

As previously discussed, the rate of online grooming and communication with children grows year on year, which could question the effectiveness of such legislation. The legislation, however, is only as effective as the reporting mechanisms in place to bring such cases of illegal communication with children to the attention of the relevant authorities.

In December 2020, the UK Government released a response to the Online Harms White Paper. The response presented by the Secretary of State for Digital, Culture, Media and Sport and the Secretary for the Home Department. The Online Harms White Paper put forward ambitious plans by government to bring into force a new way of placing greater accountability and oversight on tech companies with a government agenda of moving away from self-regulation by. The government propose a regulatory framework that outlines the responsibilities tech companies must adhere to keep UK internet users and in particular children safe online. The focus of this framework is to mitigate the exposure to illegal content and illegal activity of malicious actors online. The white paper proposes an independent regulatory body, which will set out clear online safety standards as well as holding new enforcement powers.

In part of the response by government to the white paper, government outline several cases for their support of the white paper and the measures it proposes to regulation of online content and required change to the UK legal system. The response goes on further to outline in their case for support of new measures in sections 8, 10, 11 and 13, which discuss the alarming statistics and apart risks to children online. It is therefore, posited by this research that with the support and enforcement by such government initiatives that technology companies must look to bring into being far- and wide-reaching protection mechanisms that keep young children and vulnerable adults safe online. (Dowden & Patel, 2020)

Recent figures for the use of social media by children showed that Facebook was the most popular platform to be used during 2019 for 12- to 15-year-olds. Other platforms such as *SnapChat* and *Instagram* also proved very popular. *Facebook* displayed 69% of respondents were using the platform whilst 62% of respondents used another popular messaging application called *WhatsApp*.

However, in the same report from *Statista.com*, (2020) the percentage of children aged 12 to 15 years using social media accounts has dropped from a peak of 80% in 2012 to 70% in 2018. This may be a direct result of the major social networks implanting a minimum age policy within their terms of service. (statista.com, 2020). According to *Ofcom*, 5% of children in the 5- to 7 years age bracket had use of a social media account in 2018. (Ofcom, 2020)

A list of current social media platforms compiled by so*cial-media.co.uk* is detailed in Table 3 below. Also detailed in the table is the UK and global number of users of each platform. The list in Table 3 is not comprehensive but highlights a selection of the most popular social media platforms.

Table 3: List of current social media platforms in 2020. Reproduced from (social-media.co.uk, 2020)

Current Brand Logo	Media Plaform	Description	UK Users	Global users	Daily Usuage
f	Facebook:	A social sharing networking site that allows registered users to create posts, upload video and digital images, send text, voice and video messages. Users can interact with friends and family by accepting requests to join their friends list. Users can also microblog via use of a social wall that allows users to create buttletins or status updates about their daily lives. Friends of the users can comment upon status updates and uploads via the wall feature of the platform. There is also an instant messenger application built into the platform where user can interact privately in either a one to one or one to many interation.	35,130,000	2,417,000,000	Highest traffic occurs between 1-3pm, however more engagement can be found between 7-8pm
ask.fm	Ask.fm	Ask.fm is a free app and website that allows users to post anonymous comments and questions to a person's profile. Users are able to sign in using their account details from other popular platforms such as Facebook or Twitter. The application makes it easy for users to share their content across a variety of social media platforms such as Facebook, Twitter and Tumblr. Users are	Not Available	100,000,000	Usage at its peak was around 12m per day but has dwindled over the last few years to 1m per day in 2020

		able to cross-post questions between their Facebook walls and Twitter Feeds. Facebook also allows operation of the Ask.fm application from within the Facebook Application			
	YouTube:	Is a website used for video uploading and viewing. YouTube allows user to upload and share video content that can be rated according to the viewers enjoyment or relevance of the video. The platform also allows users/viewers to comment on videos via the channel feature.	23,000,000	1,900,000,000	1 billion hours of Youtube are watched daily
	WhatsApp:	Launched in 2009, WhatsApp is a text and voice messaging service. The application reads the existing contacts within the user's device and creates those contacts within the application automatically. However, users can create new contacts within the application should they chose to.	Not Available	1,500,000,000	55 billion messages are sent each day on the platform
	Instagram:	Instagram is a free photo and video sharing app available on iPhone and Android. The platform allows users to upload both digital photographs/images and video. Whilst the platform has facility for users to share with friends, there is also the facility for followers of your Instagram to view user's images and comment.	14,000,000	1,000,000,000	95 million posts are shared each day on Instagram
	Twitter:	Initially created in 2006 twitter is a Micro-blogging and social media platform that allows users to post and interact with messages known as "Tweets". Users of the platform have the ability to post, like and retweet.	13,000,000	883,000,000	Year-on-Year the total ad engagement rate was up to 91%
	Tik Tok:	App that allows users to create and share short 15 second videos.	Not Available	800000000	The young demographic of Tik Tok users spend almost an hour on the application every day.
in	LinkedIn:	B2B platform for networking professionally.	20,000,000	590,000,000	More than 50% of all social traffic to B2B sites and blogs comes from LinkedIn
	Snapchat:	Send images and videos with a short life span over an app.	16,200,000	600,000,000	528,000 Snaps are shared every minute
t	Tumblr:	A popular microblogging platform used to broadcast messages.	9,000,000	550,000,000	45% of Tumblr's audience is people under the age of 35
	Reddit:	An entertainment, social news and social networking website.	6,600,000	330,000,000	The average time a Reddit user spends on the site is 16 minutes
S	Skype:	Telecommunications application that provides video and voice calls via the Internet.	Not Available	300000000	The mobile app has been downloaded over 1 billion times
	Pinterest:	A popular photo sharing website.	10,300,000	266,000,000	50% of users have made a purchase after seeing a Promoted Pin on their feed
	Flickr:	An image hosting website used to showcase photography work.	Not Available	112,000,000	On a high traffic day, Flickr users upload around 25 million photos
	Vimeo:	A video uploading and sharing website.	412,085	80,000,000	Vimeo gets 715 million monthly video views

F	Foursquare:	A local search and discovery app.	Not Available	60,000,000	50 million brands have used Foursquare at least once
M	Medium:	An online publishing platform.	Not Available	60,000,000	70% of Medium users are under the age of 50
F6S	F6S:	The largest platform for founders in the world.	Not Available	2,000,000	Currently there are over 22,000 start-ups registered on F6S
cb	Crunchbase:	A platform that stores information on private and public businesses.		140,000	51.7% of visits come from the US, while 2.7% of visits come from the UK

Whilst some of the platforms listed in Table 3 are business or start-up focused, most of the platforms do have a messaging service built into the platform or their primary function is social interaction.

Recent national newspaper articles have highlighted the number of child users of social media platforms and that some, under the minimum age required to use the platforms actively lie about their age to gain access. In 2012 *zdnet.com* ran an article that claimed that 38% of *Facebook* users were under the age of 13. This figure equated to an estimated 7.5 million *Facebook* users and in addition to these, 5 million users were under the age of 13. (zdnet.com, 2012)

It is therefore apparent here that this is not a new phenomenon nor have measures such as minimum age polices been effective in the prevention of child interaction with social media platforms.

There are a mixture of guidance and applications available for the protection of children online. Guidance places responsibility upon the social media provider in applying best practice approaches to the deployment of social media services. Law does not bind this guidance and therefore it is the decision of the provider as to whether such best practice is followed. This research has found, and highlighted recent cases of grooming online using social media platforms and therefore questions the effectiveness of the response to and implementation of the guidance set out in the following sections or this research.

Other guidance is directed towards parents, guardians, educators, and other agencies that are responsible for the safeguarding, moral, legal, and ethical wellbeing of children. Whilst the onus is placed upon such adults in positions of responsibility, there is a theme within the guidance that such responsible adults have the necessary digital skills and awareness to monitor, implement and maintain such approaches as set out by the various guidance available.

Child protection applications are, in the main, reactionary pieces of software that offer protection against children visiting unsavoury web domains, that present content such as pornography and violence. In essence, such applications build a whitelist of web sites that children can visit and blocking those sites deemed to fall into pre-defined categories. In other applications, a reporting element that reports upon events after the fact. Given the speed at which online grooming takes place (minutes in some instances as this research has found) this approach is often too late.

The question must be raised relating to the level of digital literacy of the adults expected to manage such protection applications and service a suite of devices and platforms a child may have access to. The onus to place responsibility on parents and other agencies who may not have the level of competency or knowledge required to manage the digital presence of children on the internet presents shortfalls in the level of protection provided. This, especially in a dynamic, constantly shifting technology landscape where “the next big thing”, “killer platform” or “killer application” can emerge at any given time.

Figure 3 below from Office for National Statistics (ONS) displays the most popular activities of adult internet users in 2018. Whilst there is a broad range of activities detailed in the graph ranging from sending/receiving email to making health care appointments, there is no mention of safety or security of self or family members online.

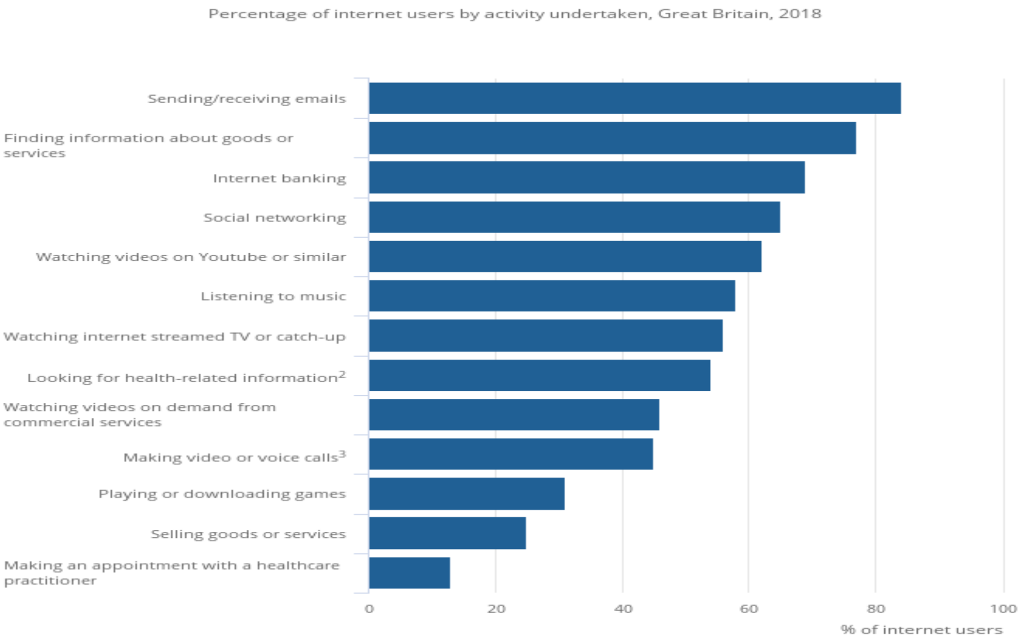


Figure 3: Percentage of internet users by activity undertaken, Great Britain, 2018. Reproduced from (ONS,2020)

In 2015 the *UK Council for Child Internet Safety* (UKCCIS) published a guidance / code of practice for the providers of Social Media platforms and services to follow to ensure the protection of child users of such platforms.

In section 4 of the guidance "Child Sexual Abuse Content or Illegal Contact" the guidance outlines how social media platforms should deal with child sexual abuse of contact by means of a best practice approach.

The guidance explains to providers that:

> "To a child sex offender, your platform represents an opportunity to gain virtual access to children, to sexually exploit them and/or to share child sexual abuse content with others. You therefore have a vital role to play in protecting your users."

And

> "To do this you must have the dedicated resources to detect and prevent child sexual abuse content and child sexual exploitation."

The guidance outlines 4 main areas of good practice to follow:

1) Give your users a standardised function for them to report child sexual abuse content and illegal sexual contact.
2) Have a specialist team, who are themselves supported, to review these reports.
3) Escalate reports of child sexual abuse content and illegal sexual contact to the appropriate channel for investigation.
4) Tell users how they can report child sexual abuse content or illegal sexual contact directly to the relevant authorities, and/or where to obtain further advice.

(UKCCIS, 2015)

In terms of Child Sex Abuse Content (imagery containing child pornography) being detected or the provider being alerted to the presence of such material, the provider is advised to report the content to the IWF whereupon the IWF will issue a "Notice of Takedown" which will request the removal of any content deemed as inappropriate that is hosted within the UK to be taken down.

Where a case of Illegal Sexual Contact Online is detected, the provider must report such content to the *Nation Crime Agency's* body responsible for the protection of children – *Child Exploitation and Online Protection* (CEOP) Command. (UKCCIS, 2015)

The apparent issue with this guidance is that there is no direct link to any legal framework within which social media providers must adhere to. The onus is placed on an approach of trust in the providers "doing the right thing" in not only detecting but reporting illegal events related to children.

Such detection and reporting mechanisms are often after the fact, where an offence may have taken place and therefore there is a lack of any autonomous real-time detection solutions being put forward nor is there any initiative for such real-time detection systems.

However, in a later guidance *UKCCIS* does outline a framework for the education and awareness of users. The guidance asks providers to consider three key questions:

- What do parents know?

  However, according to Ofcom (2015) 75% of parents had received some form of information about keeping their child safe online and managing the risks. 53% of parents had received information from school.
- What do children want?

  Children of young age in the sub 10 years group are more open to parental moderation of online activity whilst older children are more likely to be resistant to parental moderation and feel as though their privacy is being invaded (Ólafsson, 2014).
- Do children understand what they are being told?

  The information relayed to children has to be clear and well laid out otherwise, especially in younger children there is opportunity for a mixed message to be received.

(UKCCIS, 2017)

Table 4 below lists a selection of the current software tools available to parents with a brief narrative on their function. In each of the tools listed there is a distinct emphasis in placing responsibility upon the parent to select the "right tool" and to implement that tool effectively. It must be noted that all but one of the tools listed offer any form of real-time autonomous protection of children.

Table 4: Current Child Online Protection Applications

Protection Application	Safety Functions/Features	Limitations of the protection	Offers Real-Time Communication Protection	Offers an Automated Protection Service
Qustodio	Qustodio is a premium paid for application that will: Track Calls and Text messages for the Android Phone Platform.	Is limited in the number of mobile platforms it can be installed on. Not all features	No	No

	Allows Parent to view and monitor social network activity on platforms such as: Facebook, Twitter, Instagram, WhatsApp plus a range of other platforms. (Qustodio.com, 2020)	are supported across all the platforms it does support.		
Net Nanny	Is a premium paid for service that provisions parents with a range of tools to block access and monitor a range of applications, content, websites and social media. **Current protection offers:** Filter adult content in real-time: Block pornography Manage Screen Time Social Media Protection: blocking unwanted / questionable services. Content filtering of social media content. (Netnanny.com, 2021)	Reliance on parents setting a range of monitoring or blocking operations up on the various devices a child uses.	No	No
Norton LifeLock	Is a premium paid for service that provides parental controls. **Current protection offers:** Manage child screen time. Alerts about visits to inappropriate sites. Content and site blocking. (uk.norton.com, 2021)	Limited scope in protect whilst reliant upon the parent managing and blocking harmful and inappropriate content.	No	No
Kapersky Safe Kids	Is a premium paid for service providing parental controls. **Current protection offers:** Blocking access to adult content/Websites. Blocks harmful searches via YouTube. Management of access to games and inappropriate applications a child may access. Provides child psychology advice on online topics. (Kapersky.co.uk, 2021)	Limited scope in protect whilst reliant upon the parent managing and blocking harmful and inappropriate content.	No	No
Clean Router	Is a premium paid for service providing parental controls over internet access. **Current protection offers:**	Reliant upon the parent managing and blocking harmful and	No	No

	Ad Blocking Arts and Nudity Drugs related sites Gambling sites Guns Violence and Weapons Naturism Malicious Software Pornography Image sites (Cleanrouter.com, 2021)	inappropriate content.		
Bark	Is a premium paid for service that is multi-platform and monitors a range of applications and mediums. **Current protection offers:** Monitors over 30 different platforms Monitoring of text messages, email messages, social media platforms for harmful activity. Detects and alerts parents to harmful messages and activity. Allow parents to block access to harmful / inappropriate sites and content. (Bark.us, 2021)	No limitations of protection.	Yes – Text message monitoring	Yes
Mobicip	Is a premium paid for service for mobile devices. **Current protection offers:** Managing screen time. Remote locking of devices. Location tracking of child. Block social media applications and games. Web content filtering and content blocking. Manage video streaming sites and applications. Review browsing history. Offers child data privacy.	Reliant upon the parent managing and blocking harmful and inappropriate content.	No	No

In February of 2020 Ofcom published research in collaboration with the Information Commissioners Office (ICO) titled "Internet user's experience of online harms". The research surveyed a range of age groups and extrapolated data on user's attitudes and concerns about internet usage.

Amongst other statistics listed, the research found the following in relation to children's concerns about the internet and social media sites.

- 56% of 12–15-year-olds feel safe when on social media.
- 41% of 12–15-year-olds do not use social media for fear of online bullies.

- 78% of 12–15-year-olds are aware of how to change privacy setting but only 68% had done so.
- There was a 9% drop from 2019 to 50% of those children (12-15) who had concerns about interacting with other people on the internet.
- Girls had a higher concern about interacting with other people on the internet with 53%.
- 28% of adults had immediate concerns about unwelcome friend/follow/contact from strangers.
- Overall, 83% of adults had concerns about children online across all categories in the survey. Out of this target group the adults with children 92% had immediate concerns about child safety online.

(Ofcom, 2020)

One direct quote from a child participant in the Ofcom research (12–15-year) was:

"I worry that people will say something nasty about stuff I post online or that they will know personal information about me, like where I live or go to school. I also worry that someone will approach me and say something inappropriate or that they may not be who they say they are." (Ofcom, 2020)

2.2 State of the art in this research

Computer-mediated discourse is the communication that takes place between two or more human participants. This interaction takes place via the transmission of electronic messages transported between end devices (computers) connected to a computer network. Computer-Mediated Discourse (CMD) differs from Computer-Mediated Communications (CMC) in that it studies the use of language within such computer networked environments and utilises varying methods of discourse analysis to achieve its goal.

CMC on the other hand is, for the purpose of this research, the transmission of *text-based* electronic messages that are passed between two interlocutors using various text passing applications usually situated in two or more geographically distanced locations.

According to Herring, (1997) the study of CMC has its origins in the 1980s where research was first published by Naomi Baron in 1984 who's paper "computer-mediated communication as a force in language change" focused upon the linguistic changes that may occur because of CMC. However, earlier studies had been conducted by Johansen, et al. (1979) in their publication "Electronic Meetings: Technical Alternatives and Social Choices". This would suggest that interest in CMC had begun to form alongside the growth of the networked computing paradigm the World Wide Web as previously discussed.

The detection and identification of online predators using machine learning is nothing new, just as the problem of online grooming is not a new phenomenon and could stem back to the early development and popularity of Computer Mediated Discourse. According to Borj & Bours (2019) one of the early attempts to identify online grooming was carried out by Pendar (2007) who used K-Nearest Neighbours (KNN) and Support Vector Machine (SVM) to detect both participants in grooming discourse (predator and victim) and managed to return an F-score of 0.943.

Other researchers such as Parapar, et al. (2012) took the approach of using Linguistic Inquiry and Work Count (LWIC) as well as using TF-IDF and managed to successfully return an F-score of 0.849. Parapar, et al. posited that predatory discourse could be identified using psycholinguistic features. Related to the field of psychology, such study of discourse centres around the words that people use in conversation about their daily lives which can therefore reveal information about their psychological and social lives.

Cano Basave, et al. (2014) worked upon the identification of three online grooming stages that centred around the detection of Trust Development, Grooming, and Approach. It was also the intention to focus on how this may adapt to a wider application of detecting malicious conversations online in a more general context than that of online grooming.

Gupta, et al. (2012) also worked upon characterizing online grooming discourse by again as with other research in the field working on the stages of the grooming discourse. Gupta, et al. argue that simple detection of key words within grooming discourse may not be the most reliable approach to use and therefore the detection of two stages are posited – Relationship Forming (most prominent in the discourse) and, Conclusion Stage (arranging to meet and discuss travel plans etc).

Taking a different approach to grooming detection Macfarlane (2016) proposed an Agent Mediated Information Exchange for Online Real-Time Communications. This mediated information exchange employed an intelligent multi-agent environment that would act as a plugin to popular communication software with the goal of protecting child communications online. With an inbuilt ontology the proposed system also employed reasoning and NLP techniques.

Each agent in the exchange performed a specific role, which would automate the detection of children looking to arrange a meeting in person or discussing a location such as a cinema for instance. The system worked on escalated threat levels within the discourse taking place in real-time, and ultimately, the agent system would alert a parent or guardian of the threat of the conversation that had taken place. The work carried out by Macfarlane, K has proved a source of inspiration for this research and has given opportunity to collaborate on research relating to the combining of an Agent Mediated Exchange with NLP, ML and DNN architectures.

2.3 Characteristics of online groomers and the stages of grooming discourse

Craven, et al. (2006) postulated a definition of grooming as "*A process by which a person prepares a child, significant adults and the environment for the abuse of this child. Specific goals include gaining access to the child, gaining the child's compliance and maintaining the child's secrecy to avoid disclosure. This process serves to strengthen the offender's abusive pattern, as it may be used as a means of justifying or denying their actions.*"

As previously discussed in this research, the problem of online grooming displays a growing trend year on year, coupled with the continued upsurge in child interaction with technology and a plethora of social media platforms that facilitate online presence and interaction of children online. Online groomers, as suggested by Whittle, et al. (2013) are not, in the main, a homogenous group of perpetrators and therefore puts forward that in the main they are a heterogeneous group based on the variations in behaviour, offender personality, duration and intensity of the groom, and style of groom.

Whittle, et al. (2013) also discuss variations in time taken to complete the stages of the groom. This can often be related to how confident or comfortable the victim (in this instance a young person) feels during the discourse.

Time or duration of the groom can vary greatly and early analysis in this research found that this proved to be the case. Some cases of the online grooming involved a prolonged interaction with the intended victim whilst in other cases there were just 100 messages passed between predator and victim with the predator seeking to arrange physical meeting with the victim.

Whittle, et al. (2013) posit there are pre-existing models that groomers generally follow with the goal of meeting a child or young person for sex. In one such model outlined in Table 5 below, O'Connell (2003) posits 5 stages that online groomers follow to achieve their goal of meeting a child or young person for sex.

Table 5: Stages of online grooming. Reproduced from (O'Connel, 2003)

Stage 1: Friendship forming
Stage 2: Relationship forming
Stage 3: Risk Assessment
Stage 4: Exclusivity
Stage 5: Sexual

In the Friendship Forming stage the groomer establishes initial contact with the child. Groomers may request pictures and other information about the child to establish that it is indeed a child conversed with and possibly gain other information about the child such as their age. This aids the groomer in establishing whether the victim fits with their predilections.

Relationship forming looks to extend the previous stage of the groom and may engage the victim in discussing aspects of their life such as friends, family, or school for instance. The goal here is convince the child that the groomer is their best friend.

The Risk Assessment stage sees the groomer request information about the victim's location in the home, where they are using their mobile device or computer, are there any adults, siblings or significant others close by or have access to the device the victim is engaged in the conversation with the groomer. At this stage, the groomers' goal is to establish the likelihood of detection.

In the Exclusivity stage the groomer (based on the previous risk assessment) moves the conversation along and reinforces the concept of being best friends through building trust and conveying that the victim can talk to them about anything. The groomer also looks to gauge the level of trust the victim has in them and reassures the victim they can trust them implicitly. This stage sets the scene for the groom to move on into the Sexual stage.

The final stage employed by the groomer is usually introduced with questions about their anatomical self, or whether the victim has engaged in sexual acts. This introduction of sexual conversation can be introduced due to the level of trust the groomer has built up with the victim through such understanding that the victim can talk to them about anything. (O'Connell, 2003)

Another model devised in 2016 by Lorenzo-Dus, et al. proposed the first empirical model of Online Grooming Discourse (OGDM). Again, as with O'Connell's model there are stages to the grooming process but in this instance as outlined in Table 6 below, there are 6 interleaved stages that culminate in the attempt to physically meet the victim. The OGDM model built on the offline grooming model of luring communication proposed by (Olson, et al., 2007).

Table 6: Interleaved stages of online grooming. (Lorenzo-Dus & Izura, 2016)

Access
Deceptive Trust Development
Sexual Gratification
Compliance Test
Isolation

Figure 4 below displays the overlapping processes a groomer engages in with the goal of meeting a potential victim for sex.

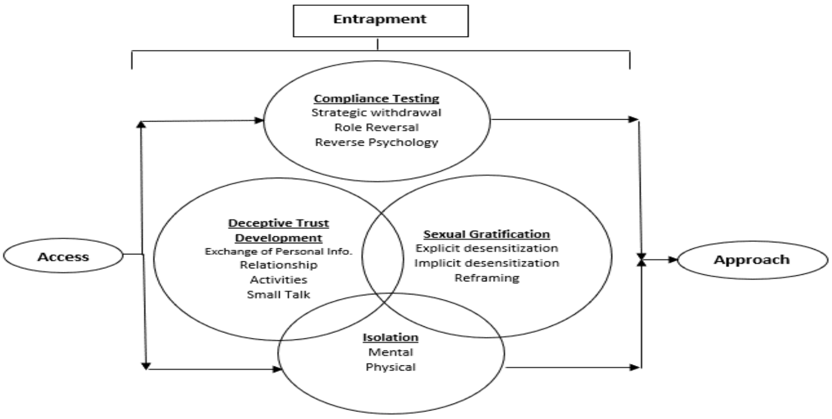


Figure 4: Online Grooming Discourse Model. Reproduced from (Lorenzo-Dus et al., 2016).

Access in this case is the initial contact between groomers and potential target and victim so in effect accessing a digital medium and making contact. The approach process is the goal of the grooming activity which results in the physical meeting with the victim or target.

Deceptive Trust Development masks the true goal of the interaction in which the groomer wants their victim to take part in sexual activities. This is conducted through building trust-based relationships where friendship or love is emphasised. There are five stages identified within this phase – Exchange of Personal Information, Relationship, Activities, Compliments, Small Talk.

Sexual Gratification relates to the groomer attempting to engage the potential victim in sexual activity. This could be, as Lorenzo-Dus, et al. suggest, a preparatory approach with the goal of sexual interaction offline (physical meeting). This could be anatomical references (genitalia), explicit sexual acts, or sexual topics within the conversation. This may also involve discussion of sexual acts between the groomer and the victim.

Compliance Test phase involves discussion that tests for the potential participation of the victim engaging in potential sexual acts proposed to them. As part of the phase of the grooming conversation the groomer may engage in interrogative discourse where the groomer may look to establish the age of the victim and thus establish the likelihood of the victim taking in future sexual acts. This phase also comprises of three strategies that according to Lorenzo-Dus et al. use “reverse psychology” where the groomer challenges the victim, “strategic withdrawal” where the groomer places the victim in control of the relationship and “role reversal” where a cause for concern of safety is conveyed.

Isolation phase involves the groomer looking to distance the victim from the meaningful people in their life such as family members – mother or father for instance. The groomer also reinforces the bond, strength, and secrecy of the relationship between groomer and victim. The groomer may, through the on-going building of trust coerce the victim into disclosing in-depth information about themselves and therefore looking to succeed in becoming a confidant. This provides a vehicle with which to get the victim to sever emotional ties with those around them such as parents. Culpeper, (2011) defined such discourse as “impoliteness talk” which involves criticism targeted at those the victim may have meaningful relationships with and as such mentally isolate the victim (mental isolation).

Both models proposed have similarities, which aid in the profiling and recognition of a grooming conversation. The OGDM model of Lorenzo-Dus et al. is used in this research to establish the basis for proposing the identification grooming discourse and potentially identify characteristics fitting with each stage of online grooming.

2.4 Existing Chat datasets for analysis

There is a rich variety of existing corpora available for analysis using NLP, focused primarily upon classification and sentiment analysis. The available datasets have been collated from product, website, and movie reviews to chat corpora.

The corpora discussed below is not an exhaustive list of corpora, but is corpora considered during this research for suitability and informing corpora selection.

The Brown Corpus of Standard American English is the first modern general corpus that was computer readable. The corpus compiled by Francis & Kudera (1967) comprised of one million words printed in American English texts in 1961. The Brown corpus laid the way for the compilation of other corpora which went some way to match the work of Francis and Kudera. The corpus consists of 15 different categories and 500 texts with an average of 2000 words per text.

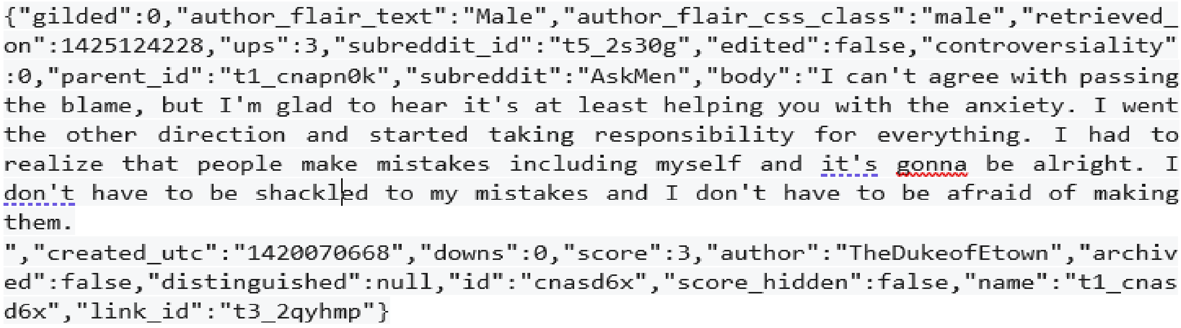
Whilst this corpus is dated, given the rate at which language evolves in digital discourse, it provides some notion to the compilation of corpora and the factors that influenced the compilation of modern corpora. (Thurlow & Mroczek, 2011)

In 2016 Lison & Tiedmann produced the improved Open Subtitles Data that looked to pre-process and improve the linguistic quality of the data in the Open Subtitles database. The Open Subtitles database contained over 3 million subtitles in more than 60 languages in 2016. The Open Subtitles corpora included 2.7 billion sentences which equated to 17.2 billion tokens, again, distributed over 60 languages.

Lison & Tiedmann's dump of the Open Subtitles database consisted of 3.36 million files that they filtered out to include only those languages that included more than 10 subcategories. The use of subtitles could prove useful for chat analysis as the subtitles emulate normal discourse ie: movie scripts that discuss that contain dialogue in a range of contexts. (Lison & Tiedmann, 2016)

The *Reddit* comment dataset compiled from *Reddit's* publicly available comment dataset by a variety of *Reddit* members who were interested in the data for NLP analysis. The available dataset is a compressed file of 250GB in size, which equates to 1.2TB total data uncompressed.

The dataset can be parsed and processed to isolate the raw chat/comment data within the file. The data is stored in JSON format within the file(s) and hence the need to parse the data to retrieve relevant chat data. A sample of the raw data can be seen below in Figure 5.



```
{"gilded":0,"author_flair_text":"Male","author_flair_css_class":"male","retrieved_
on":1425124228,"ups":3,"subreddit_id":"t5_2s30g","edited":false,"controversiality"
:0,"parent_id":"t1_cnapn0k","subreddit":"AskMen","body":"I can't agree with passing
the blame, but I'm glad to hear it's at least helping you with the anxiety. I went
the other direction and started taking responsibility for everything. I had to
realize that people make mistakes including myself and it's gonna be alright. I
don't have to be shackled to my mistakes and I don't have to be afraid of making
them.
","created_utc":"1420070668","downs":0,"score":3,"author":"TheDukeofEtown","archiv
ed":false,"distinguished":null,"id":"cnasd6x","score_hidden":false,"name":"t1_cnas
d6x","link_id":"t3_2qyhmp"}
```

Figure 5: Reddit Comment Dataset Sample Post

Twitter archives are freely available and can be streamed directly from *Twitter* using the Twitter API. Developers and researchers are able to access/stream tweets related to a particular topic or event for analysis. This can enable the real-time sentiment analysis of twitter data which could provide public opinion of a topic, celebrity or current event. (twitter.com, 2021)

For analysis of historic *Twitter* data, the Internet Archive - archive.org makes historical twitter data freely available for download and analysis. The archive files are available in JSON format and have a date range from 2011 to 2021. The twitter archives vary in size; however, a typical archive is in the tens of gigabytes range. A review of the twitter archives is discussed in chapter 5.

The *Westbury Labs* freely available *Usenet* corpus is a collection of public *Usenet* postings collected between 2005 and 2011. The corpus covers 47,860 English Language news groups. The available corpus has been pre-processed to some degree where the creators of the corpus have made efforts to minimise the inclusion of non-English words, non-words, and null entries. Further pre-processing has made efforts to anonymise the data, where email addresses, URLs and News URLs were replaced with substitute text. The corpus contains over 30 billion words and is 38gb in size in original format, however this was later reduced with *Westbury Labs'* redundant text removal algorithm which reduced the corpus to 7 billion words and 8gb in size. (Shaoul & Westbury, 2019)

The *IMDb* review dataset is a freely available dataset used in binary sentiment classification. The dataset is a prelabelled dataset made up of movie reviews that are highly polarised. The dataset is available in three files 25,000 record labelled polar movie reviews for training, 25,000 records for testing. The dataset can be used in the training of supervised machine, and deep learning models to detect the sentiment of a given text or document. The dataset has proved to be popular in the evaluation of machine learning models and deep learning architectures by researchers interested in performing sentiment analysis.

The website *pervertedjustice.com* collects chat room data from individuals who act as agent provocateur in chat rooms with the purpose of engaging perpetrators of online grooming in grooming discourse. Posing as children, adults will engage with groomers and, in most cases, continue the conversation through the various stages of grooming discourse to its conclusion and thus the arrangement of an offline meeting for sex. The data gathered by pervertedjustice.com is passed to the relevant law enforcement and in many cases has led to the criminal conviction of the groomer.

There are over 600 cases available in the *Perverted-Justice.com* datacentre, which are available by web scraping the website or by requesting access to the background data centre.

Numerous researchers have used the P*erverted-Justice* data for coding and analysis of online grooming discourse through to testing of deep learning approaches in the detection of grooming stages related to the models of online grooming previously discussed. A critical analysis of the suitability and performance of this grooming data is discussed in chapter 5.

2.5 Software used to gather data about conversation.

This section will investigate a range of available tools that can be used to analyse and visualise the characteristics of chat corpora. The aim here, is to establish the use of a suitable tool that can be used to analyse predatory discourse and highlight otherwise unknown characteristics of the data.

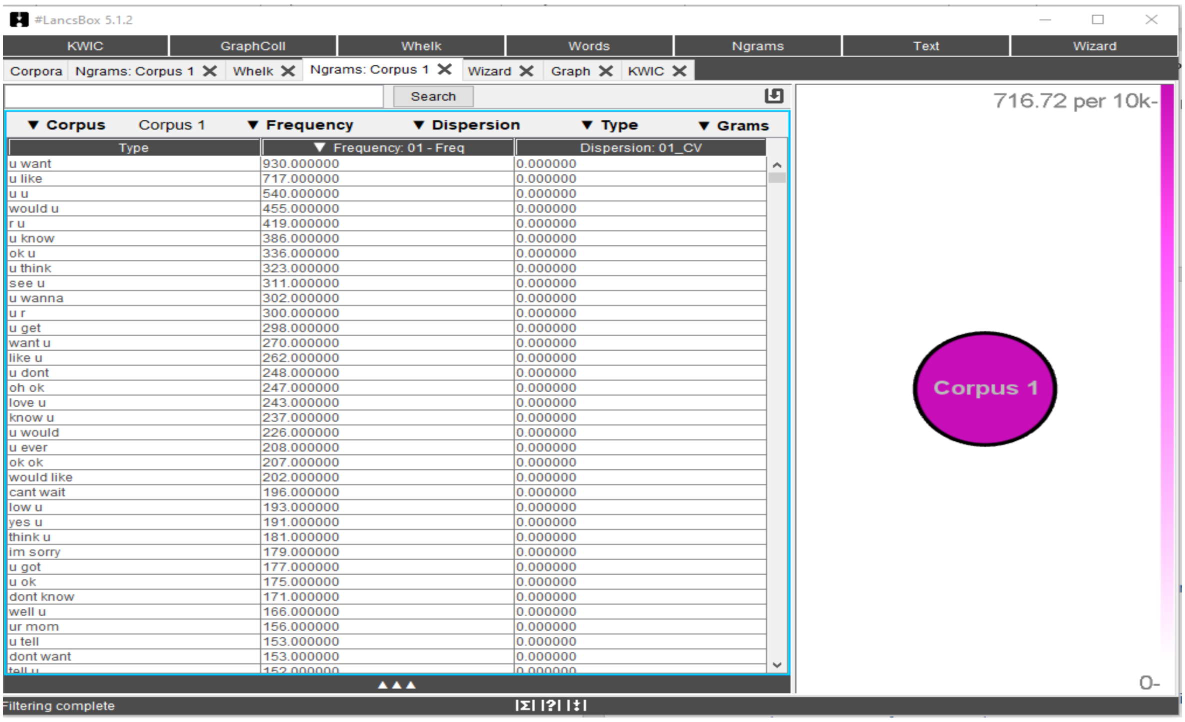
#LancsBox is a corpus analysis tool developed at Lancaster University. The tool allows you to work with your own corpus data in any language. The tool uses cutting edge technology incorporating sophisticated statistical approaches that provide the following features and reports.

Some of the main features of the tool are:

- Visualisation of language data.
- Analyse data in any language.
- Automatically annotates data for part-of-speech (POS).
- Identify collocations of words in a corpus.
- Perform pre-processing of corpus data (stopword, lemmatisation, stemming).
- Provides frequency distribution statistics.
- Word Counts of corpus data.
- Provides statistical analysis of Ngram types.
- KWIC – a tool for the generation of a list of all instances of a term within the chosen corpus. This tool presents the list in the form of a concordance.
- GraphColl - a tool that identifies the collocations of words in the corpus and presents them in table, graph or network format.
- The Whelk tool provides information about how the search term is distributed across corpus files.
- Words – a tool that performs in-depth analysis of frequencies of types, lemmas and POS categories. The tool also performs a comparison of corpora using keywords.
- Ngrams – a tool the performs in-depth analysis of frequencies of ngram types, lemmas and POS categories. The tool also performs a comparison of corpora using key ngram technique.
- Text – a tool that performs an in-depth insight about the context in which a word or phrase is used.

(Brezina, et al., 2020)

Figure 6 below is a sample screen shot of the Ngrams tool returning the statistics of the bigrams within the predator corpus used in this research.



Type	▼ Frequency: 01 - Freq	Dispersion: 01_CV
u want	930.000000	0.000000
u like	717.000000	0.000000
u u	540.000000	0.000000
would u	455.000000	0.000000
r u	419.000000	0.000000
u know	386.000000	0.000000
ok u	336.000000	0.000000
u think	323.000000	0.000000
see u	311.000000	0.000000
u wanna	302.000000	0.000000
u r	300.000000	0.000000
u get	298.000000	0.000000
want u	270.000000	0.000000
like u	262.000000	0.000000
u dont	248.000000	0.000000
oh ok	247.000000	0.000000
love u	243.000000	0.000000
know u	237.000000	0.000000
u would	226.000000	0.000000
u ever	208.000000	0.000000
ok ok	207.000000	0.000000
would like	202.000000	0.000000
cant wait	196.000000	0.000000
low u	193.000000	0.000000
yes u	191.000000	0.000000
think u	181.000000	0.000000
im sorry	179.000000	0.000000
u got	177.000000	0.000000
u ok	175.000000	0.000000
dont know	171.000000	0.000000
well u	166.000000	0.000000
ur mom	156.000000	0.000000
u tell	153.000000	0.000000
dont want	153.000000	0.000000
tell u	152.000000	0.000000

Figure 6: #LancsBox Ngram Tool

#LancsBox also produces feature rich reports and visualisations about corpus data such as the data displayed in table below. Table 7 displays collocates for the search term of “mom” within the predator data. The word “mom” is substituted for the word “mum” or “mother” in the search criterion as the corpus being analysed is based in the United States (US) and the common term used to reference a child’s mother is the term “mom” which was apparent in the initial analysis of the predator data discussed later in chapter 5.

Table 7: Collocates of the search term “mom” in Corpus 1

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
1	R	u	291	291	15302
2	L	ur	184	184	2039
3	R	ok	89	89	3802
4	R	home	68	68	488
5	R	oh	62	62	1293
6	R	cool	53	53	1294
7	R	dad	52	52	247
8	R	get	45	45	2166
9	L	low	44	44	2861
10	R	work	41	41	620

Table 7 provides the top ten collocates of the search term with a full comprehensive listing of collocates detailed in the appendix of the report #LancsBox produces. From this initial list it is easy to identify potential trigram search terms of “ur mom home” where “ur” precedes the search term (L = Left of search term) and “home” follows the search term (R = Right of the search term).

This trigram could form part of an interrogative communication with the intended victim and form part of The Deceptive Trust Development phase of the grooming model posited by Lorenzo Dus et al. (2016).

Another informative feature of #LancsBox is the ability to graphically represent collocates and provide a narrative on the collocates within the plot. Figure 7 displays the collocation network for the search terms “mom” and “dad” as above. The full analysis for the search terms “mom” and “dad” can be seen in Appendix 7.

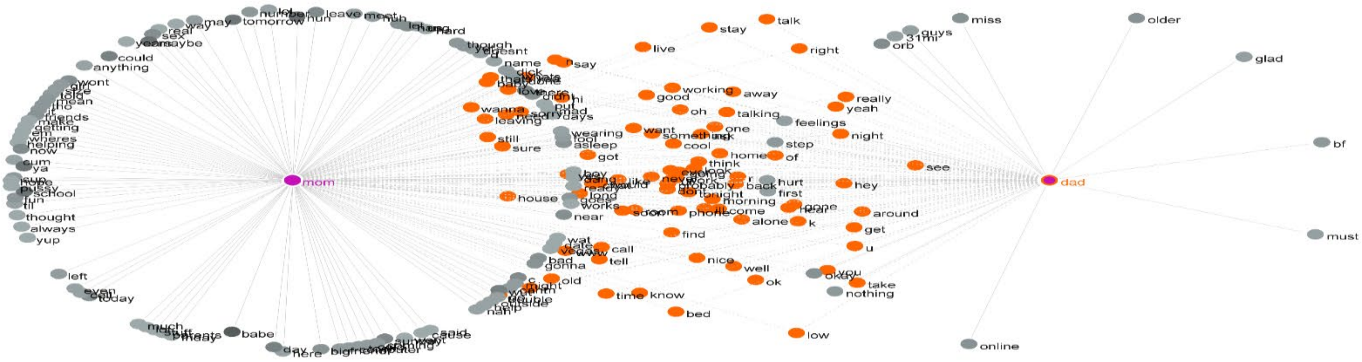
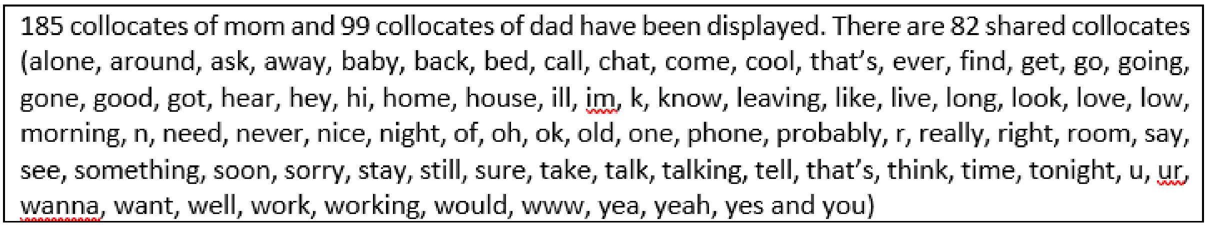


Figure 7: Collocation network:" mom" and "dad"

Figure 8 displays the narrative of both terms and their collocations as well as any shared collocates within the corpus. #LancsBox also provides a full file of collocates and statistics for further analysis.

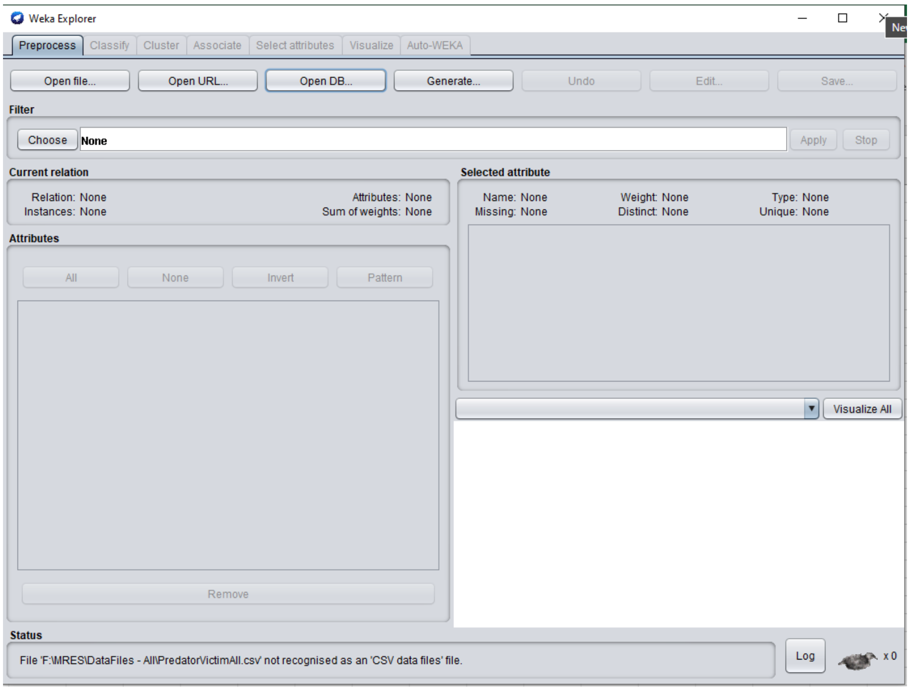


185 collocates of mom and 99 collocates of dad have been displayed. There are 82 shared collocates (alone, around, ask, away, baby, back, bed, call, chat, come, cool, that's, ever, find, get, go, going, gone, good, got, hear, hey, hi, home, house, ill, im, k, know, leaving, like, live, long, look, love, low, morning, n, need, never, nice, night, of, oh, ok, old, one, phone, probably, r, really, right, room, say, see, something, soon, sorry, stay, still, sure, take, talk, talking, tell, that's, think, time, tonight, u, ur, wanna, want, well, work, working, would, www, yea, yeah, yes and you)

Figure 8: Collocation narrative and collocation samples

Whilst #LancsBox can report and highlight such informative features it is unable to detect or predict the instances or phases of conversation. However, such features can form the basis for further investigation, analysis, and the development of novel approaches for the prediction/detection of predatory discourse. (Brezina, et al., 2020)

Weka is a well-established machine learning software tool used for analysis and classification of text data. The Weka workbench/explorer offers a range of tools for pre-processing of data through to analysis and the training of machine learning models. Weka also includes tools for data mining such as classification, clustering, attribute selection, association rule mining, and regression. A range of visualisations is available in Weka workbench which can aid in better understanding and gleaning new information about the data may have gone unnoticed before. One of the unique features of Weka is its ability to process both text data (from raw text or URL) and SQL data alike. Figure 9 displays the typical Weka interface for processing and analysing data.



Filter: None

Relation: None
Instances: None
Attributes: None
Sum of weights: None

Name: None
Missing: None
Weight: None
Distinct: None
Type: None
Unique: None

File 'F:\MRES\DataFiles - All\PredatorVictimAll.csv' not recognised as an 'CSV data files' file.

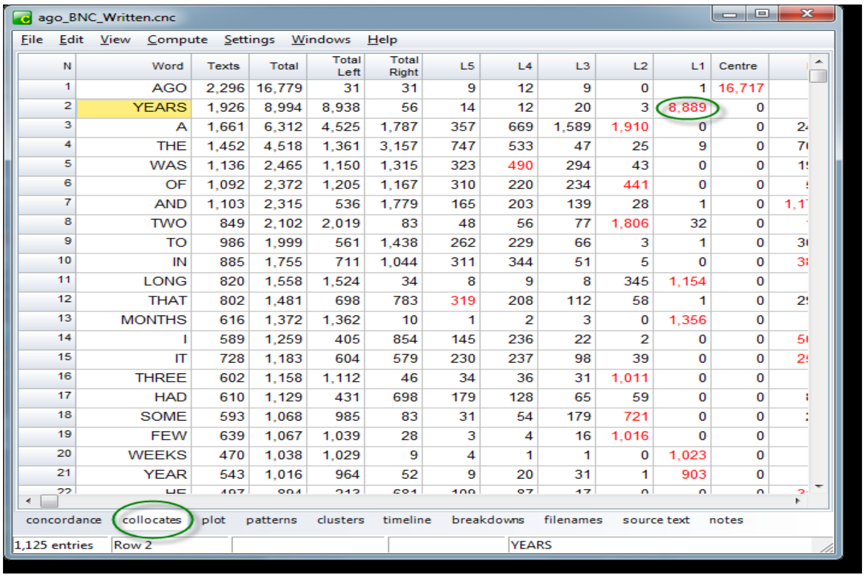
Figure 9: Weka Explorer

(Frank, et al., 2016)

Wordsmith tools is a proprietary software used for the analysis of texts. The tool offers a limited range of functions related to collocation, concordance, word lists and frequencies, lemmatising statistics, and counts of ngrams.

The tool also offers keyword statistics offering frequency of terms within a source text(s) and how the frequency compares to its frequency in a reference corpus. There is functionality to plot distribution of keywords within a corpus which is centred around patterns, clusters, and timelines.

Figure 10 below displays the statistics returned by Wordsmith on collocations. Due to the proprietary nature of Wordsmith it has not been possible within this research to test the tool and the functions offered.



N	Word	Texts	Total	Total Left	Total Right	L5	L4	L3	L2	L1	Centre
1	AGO	2,296	16,779	31	31	9	12	9	0	1	16,717
2	YEARS	1,926	8,994	8,938	56	14	12	20	3	8,889	0
3	A	1,661	6,312	4,525	1,787	357	669	1,589	1,910	0	0
4	THE	1,452	4,518	1,361	3,157	747	533	47	25	9	0
5	WAS	1,136	2,465	1,150	1,315	323	490	294	43	0	0
6	OF	1,092	2,372	1,205	1,167	310	220	234	441	0	0
7	AND	1,103	2,315	536	1,779	165	203	139	28	1	0
8	TWO	849	2,102	2,019	83	48	56	77	1,806	32	0
9	TO	986	1,999	561	1,438	262	229	66	3	1	0
10	IN	885	1,755	711	1,044	311	344	51	5	0	0
11	LONG	820	1,558	1,524	34	8	9	8	345	1,154	0
12	THAT	802	1,481	698	783	319	208	112	58	1	0
13	MONTHS	616	1,372	1,362	10	1	2	3	0	1,356	0
14	I	589	1,259	405	854	145	236	22	2	0	0
15	IT	728	1,183	604	579	230	237	98	39	0	0
16	THREE	602	1,158	1,112	46	34	36	31	1,011	0	0
17	HAD	610	1,129	431	698	179	128	65	59	0	0
18	SOME	593	1,068	985	83	31	54	179	721	0	0
19	FEW	639	1,067	1,039	28	3	4	16	1,016	0	0
20	WEEKS	470	1,038	1,029	9	4	1	1	0	1,023	0
21	YEAR	543	1,016	964	52	9	20	31	1	903	0

1,125 entries | Row 2 | YEARS

Figure 10: Wordsmith Tools Collocation Analysis.

2.6 Data storage – pre-processing, processing, and analysis

To store any form of digital discourse there needs to be some form of data storage platform to store and archive the data and any related meta data. From the early of origins of the file store to the evolution of RDBMS and a standardised way of doing things with Structured Query Language (SQL) or, of late, the unstructured data stores such as NoSQL using document stores, the ability to collect data in a plethora of different guises has been in existence almost from the birth of the stored program concept of storing both data and instructions in computer memory. (Codd, (1970); Turing, (1936))

RDBMS are a well-established technology that were first proposed by Edgar Codd in 1970. His paper "A Relational Model of Data for Large Shared Data Banks" Codd posited an approach that would move away from the heterogeneous nature of file stores where the format of file stores were often bespoke affairs only useful to the system they were written for. Codd devised a relational model that would reduce or eradicate the storage of redundant data and by having a schema that described the data and the domain in which the data existed as well as the relations between data. Codd's approach allowed the data store to sit outside of a computer system as a separate entity with programs/applications connecting to the data store via a RDBMS. This would enable multiple programs or systems to have a uniform view of data and therefore increase the utility of the data whilst reducing redundant/repeated data. (Codd, 1970)

Through the design and use of database schema and normalisation approaches, data relations could be formed in such a way that the performance of the RDBMS and applications accessing the data were improved. A schema typically identifies the "Entities" that exist within a domain and the "attributes" that describe the entity.

Data within an RDBMS is typically stored within tables that have some relation between them achieved by the use of key fields (Primary (PK) or Foreign (FK)) and relations that can be of:

- one to one relationship
- one to many relationship
- many to many relationship

For the purposes of this research, three such entities have been identified:

- **Predator** – malicious interlocutor intent on meeting a child for sex.
- **Victim –** young person/child engaged in a grooming conversation.
- **Case –** contains the discourse that takes place between Predator and Victim.

The assignment of attributes that describe each entity will aid in preserving the anonymity of Predator and Victim by storing a numerical value (ID) in the data table. Figure 11 displays a typical database schema and the relations between entities/tables.

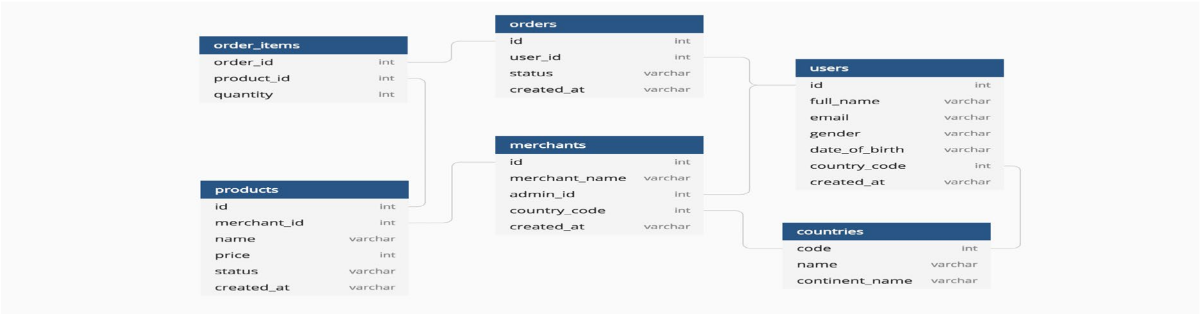


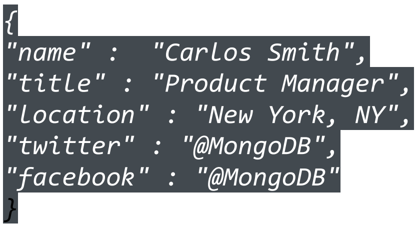
Figure 11: Example Database Schema

The sample schema in Figure 11 displays the typical format of a schema with entities and their attributes. Note that attributes are of a type which represent the type data being stored within the table ie.: Varchar (text of variable character length) or Int (an integer number).

MongoDB is a popular proprietary / open-source document database that stores data in what are referred to as JSON-like documents. This is a move away from traditional RDBMS and embraces the notion of NoSQL where queries on the data stored in its JSON-like documents are constructed in JSON rather than traditional SQL. Document databases offer a dynamic flexible approach to data storage and retrieval.

Documents work with fields of data that are comparable to the columns of a traditional RDBMS using SQL. Highly scalable, MongoDB can be distributed across a number of machines to create interconnected systems that scale out in not only data size but also performance in speed.

A typical JSON document resembles the document displayed in Figure 12 below where fields and data for a record are contained as one.



```
{
"name" :  "Carlos Smith",
"title" : "Product Manager",
"location" : "New York, NY",
"twitter" : "@MongoDB",
"facebook" : "@MongoDB"
}
```

Figure 12: JSON Document Example for Contact Details. Reproduced from (MongoDB, 2021)

Apache Cassandra is another NoSQL data store that boasts fantastic performance when compared to other NoSQL alternatives. This highly scalable datastore can see deployments ranging in the petabyte range distributed over tens of thousands of nodes working concurrently. Cassandra is highly fault tolerant and able to recover from failure not only at node level but also at the datacentre level. This achieved through smart replication and distribution amongst connected nodes where failed node data is recovered or redistributed to other connected nodes in the system. Cassandra’s scale is elastic, as nodes can be added or taken away (scale up or scale down) without any service interruption.

The benefit of using Cassandra is the protection or persistence of data in environments where data integrity and availability are key requirements. Popular community-based website Reddit employs Cassandra in storing community comments. (Cassandra, 2021)

Apache Hadoop is a framework that enables the processing of large data sets (Big Data) across computer clusters. Hadoop is highly scalable from single instance to clusters with nodes numbering in the thousands.

Again, like Cassandra, Hadoop is highly fault tolerant and able to distribute copies of data across nodes of the cluster and thus recover data in the event of failure of a particular node or nodes for instance.

Hadoop has a variety of modules available such as Hadoop YARN for job scheduling, and Hadoop Map Reduce for the parallel processing of Big Data analytics typically in the terabyte range. Hadoop also includes its own dedicated file system HDFS that provides high throughput to data. A drawback of running applications using Map Reduce is the distribution and iteration over data using HDFS and HDDs (Hard Disk Drives) which, as a secondary memory store can slow performance when compared to in-memory operations performed by frameworks such as Apache Spark. (Hadoop, 2021)

Apache Spark is akin to Apache Hadoop in that it handles Big Data analytics in a scalable cluster environment. Unlike Apache Hadoop, Spark performs the bulk of its operations in Random Access Memory (RAM) and is therefore able to realise performance gains of 100 x faster than the same job executed on Apache Hadoop.

Spark can interact easily with a range of languages such as Java, Scala, Python, R, and SQL. Spark is versatile in that it can run as a standalone cluster installation or on a variety of platforms including Hadoop, Apache Mesos, Kubernetes, or via cloud instances. (Spark, 2021)

The big data solutions discussed above can integrate machine learning solutions therefore performing operations in the big data range of terabytes of data which can be generated in modern social media platforms. The

2.7 Pre-processing of data

To perform NLP operations on text data it is necessary to perform a range of pre-processing or normalisation operations. Data needs to be in a consistent form before the commencement of operations and therefore a workflow of pre-processing of data needs to be followed. Figure 13 outlines a typical workflow for pre-processing; however, this is not a fixed workflow as there must be consideration of the operations on the data post pre-processing.

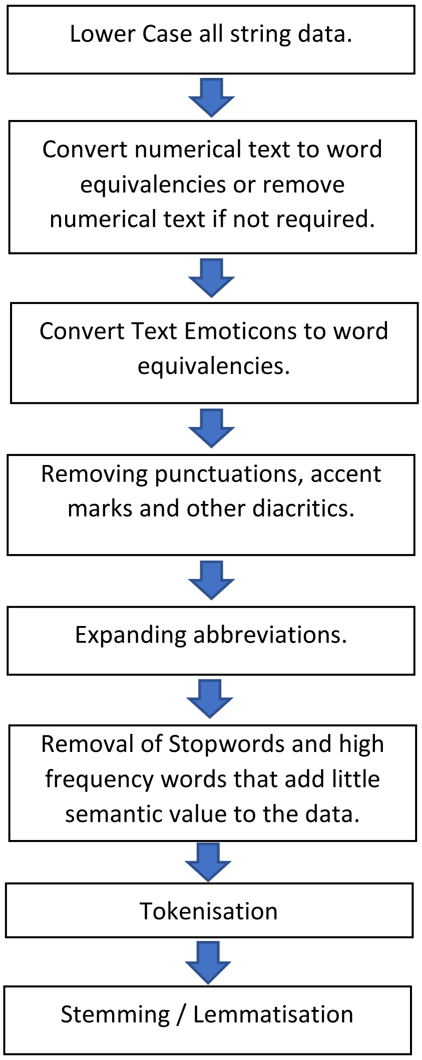

Lower Case all string data.
Convert numerical text to word equivalencies or remove numerical text if not required.
Convert Text Emoticons to word equivalencies.
Removing punctuations, accent marks and other diacritics.
Expanding abbreviations.
Removal of Stopwords and high frequency words that add little semantic value to the data.
Tokenisation
Stemming / Lemmatisation

Figure 13: NLP - Pre-Processing Workflow

Operations on text data carried out at each phase in the pre-processing workflow are as follows:

Lowercase data – provides a better consistency of data across the corpus being worked upon. In carrying out this operation machines are able to see the data as a consistent token in the corpus rather that tokens that although are the same word, to the machine they have subtle differences. In the American Standard Code for Information Interchange (ASCII) the value for the uppercase version of the letter "A" has a decimal value of 65 in the ASCII table whereas the lowercase letter "a" has a decimal value of 97 in the ASCII table. When converted to their binary equivalents for the machine to understand "A" becomes 1000001 and "a" becomes 1100001 therefore two completely different data values in the eyes of the machine. (asciitable.com, 2021)

A typical lowercase operation would therefore perform the following:

Uppercase = "Hey I am 14 Years old!!!!... :(x"

to

Lowercase = "hey I am 14 years old!!!!... :(x"

Conversion of numerical data or removal – Again, as with lowercasing of data we can ensure consistency across the data. Processing of numerical data within a corpus may take differing approaches dependent upon the nature of the analysis being carried out. In some instances, it may be acceptable to remove numerical data which holds no semantic value in the data whereas in other cases there may be the case to preserve the numerical values for mathematical analysis.

A simple example of this could relate to grooming data where the age of the participant in the online conversation could be value in detecting grooming discourse.

To put this into context we would therefore convert the following:

"hey I am 14 years old!!!!... :(x "

to

"hey I am fourteen years old!!!!... :(x"

Conversion of text emoticons to text equivalences – emoticons are a means of expressing emotion within online chat conversations. Wang & Castanon, (2015) carried out an analysis of the effects of emoticons on sentiment analysis in NLP centred around how emoticons may change the polarity scores of sentiment analysis. The results of the analysis proved that emoticons have a significant part to play in accurate sentiment analysis and therefore consideration of such must figure in the pre-processing and processing of data for sentiment analysis.

Using the text example once again the text would convert as below:

"hey I am fourteen years old!!!!... :(x"

to

"hey I am fourteen years old!!!!... sad kiss"

Remove Punctuations – removal of punctuation within texts again improves the consistency of the data. Punctation can often be misplaced or included to add emphasis in the data or just random typos during the conversation. Ek, et al., (2020) tested the affects of punctation on a variety of neural models and the results of the reserch undertaken found evidence that none of the models they tested were capable of detecting cases where the punctation was meaningful. Whilst the work of Ek, et al. continues, it can be said that in the main it is better to remove punctation from the text.

Returning to the running example the text with punctuation removed would convert as below:

“hey I am fourteen years old!!!!... sad kiss”

to

“hey I am fourteen years old sad kiss”

Remove Stopwords – Luhn, (1958) first posited the concept of stopwords in his paper on information retrieval in business systems. According to the Oxford English Dictionary stopwords are defined as “stopwords are common words that are not indexed or searchable in a computer search engine”. Stopword lists (we use English stopwords here) consist of highly common words such as ‘a’, ‘and’, ‘the’, ‘as’, ‘an’, ‘all’, ‘do’.

Whilst toolkits such as Python’s NLTK have predefined lists of stopwords there is often the need to add to such lists or create a custom list based upon analysis of a corpus and thus remove words of high frequency that may have little semantic value. An added benefit of removal of stopwords is the reduction of the amount of the data being processed and thus enhancing computation with a reduction in the space and time required, especially when considering the inclusion of approaches such as TF-IDF in NLP operations. (Ladani & Deasi, 2020)

Applying stopwording to the example text once again would result in the following:

“hey I am fourteen years old sad kiss”

to

“hey fourteen years old sad kiss”

Tokenisation – is a way of splitting texts into separate units commonly referred to as tokens and can take the form of words and charters for instance. Tokens can also include subwords if such were desired. ML and DL architectures perform well when data is presented to them at the token level. Webster & Kit, (1992) were one of the early researchers in tokenisation and later Vijayarani & Janani, (2016) were amongst a wealth of other researchers in text mining / NLP that have all discussed and tested the positive affects tokenisation can bring in the performance of ML and DL models related to a range of NLP operations. A sample list of tokeniser tools is listed below.

- Nlpdotnet Tokenizer
- Mila Tokenizer
- NLTK Word Tokenize
- TextBlob Word Tokenize
- MBSP Word Tokenize
- Pattern Word Tokenize
- Word Tokenization with Python NLTK

(Vijayarani & Janani, 2016)

Applying word level tokenisation to the example text would result in the following:

“hey I am fourteen years old sad kiss”

to

[‘hey’, ‘fourteen’, ‘years’, ‘old’, ‘sad’, ‘kiss’]

Stemming and Lemmatisation

Stemming in NLP refers to the morphological variants of a root word. The existence of a word across many documents may result in a variety of forms of the word existing i.e.: “blessing”, “blessed“.

With stemming algorithms or stemmers, as they are more commonly known, words can be reduced to their root or base thus removing the morphological affixes from a word. Therefore, in the examples above both words would become “bless”. However, the aggressive nature or quirks of the stemming algorithm may result in unexpected results and therefore can result in one of the following three cases for example:

1. The word is reduced as such it becomes a different word within a dictionary:

   caring » car

2. Over-stemming of words where different words are stemmed to the same root word thus losing their original meaning such as:

   universe
   university
   universal
   universalistic

   In this case the words would be reduced to the stem of “univers”. Therefore, this instance could be referred to as a false positive.

3. Under-stemming of words where words that stem to the same root word do not stem as expected such as:

   alumnus

   alumni

   alumnae

In this case the stemmer returns a false negative.

There are a variety of stemming algorithms in existence and three of the most common are Porter Stemmer, Lancaster Stemmer and the Snowball Stemmer which is a revised version of Porter.

Applying stemming techniques to the example text would result in the following (Porter Stemmer):

[‘hey’, ‘fourteen’, ‘years’, ‘old’, ‘sad’, ‘kiss’]

to

hey	hey
fourteen	fourteen
years	year
old	old
sad	sad
kiss	kiss

In the short example text used thus far the only word stemmed was “years”.

Lemmatisation is another approach used in the pre-processing of data and therefore looks to group words together in their canonical forms allowing them to be analysed as single entity in the word’s lemmatised form.

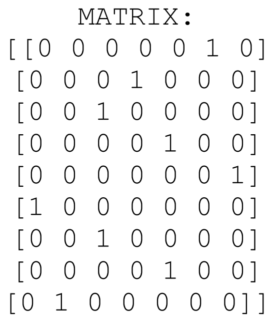
Words such as “last”, “lasted”, “lasting” are forms of the same lexeme, the word “last” is the lemma by which the words are indexed. The lexeme is the group of words that hold the same meaning. There are many researchers that have tested the effects (positive and negative) of both stemming and lemmatisation of text in the field of NLP such as Schofield & Minmo (2016) and later May, et al.(2019). The effects at times have been found to be negligable whereas other researchers have seen some improvement in the processing of the data. The effects of pre-processing techniques including stemming and lemmatisation have been tested during the course of the research.

3 Machine Learning and Deep Learning approaches used in NLP.

In NLP problems, data usually consists of a corpus of words that need to be processed in such a way that a machine or machine-learning model can understand the data. The corpus of words that require processing is referred to as Categorical Data. The process here is to transform each word in the corpus into a vector which is a binary representation or 0 and 1 and thus create a vector space based upon the size of the corpus.

For example, if we were to encode/vectorise the phrase "the house had many windows and had many doors" using One-Hot encoding with Python’s NumPy, we would return the following matrix or vector space seen below in Figure 14 which produces a matrix of words within the corpus.

In generating the matrix in Figure 14 an initial index is created for each unique word in the corpus. This would create a matrix of 9 x 9 dimensions where the number of words in the vocabulary signifies the number of dimensions created.



```
    MATRIX:
[[0  0  0  0  0  1  0]
 [0  0  0  1  0  0  0]
 [0  0  1  0  0  0  0]
 [0  0  0  0  1  0  0]
 [0  0  0  0  0  0  1]
 [1  0  0  0  0  0  0]
 [0  0  1  0  0  0  0]
 [0  0  0  0  1  0  0]
[0  1  0  0  0  0  0]]
```

Figure 14: One-Hot Encoding Matrix based on a vocabulary of 9 words.

Whilst One-Hot encoding enables a machine to process data using this representation, there is no notion of similarity captured within the process and given that unique words are encoded within the representation. Therefore, the Cosine similarity of unique words would be zero and the Euclidean distance between vectors would always be sqrt(2) which would denote that any notion of semantic information is not expressed.

Another drawback to employing One-Hot encoding is the computational cost of processing large corpora where the number of dimensions is very large. Even in a small vocabulary of 50,000 words, each word in the matrix would be represented by 49,999 zeros and a solitary one in the vector. To achieve some sense of scale in terms of memory requirement to store such a matrix for a 50,000-word vocabulary a total memory allocation of $50{,}000^2$ (2.5 billion) units of memory would be required. The temporal and spatial processing of such a matrix would be computationally expensive and therefore not an efficient approach to use in isolation.

The issue of dimensionality and dimension reduction related to large vocabularies can be addressed using approaches such as Skip-gram models or Continuous Bag of Words.

In 2013, Mikolov et al. introduced the Skip-gram model, which was an efficient approach to learning high quality vector representations of words from large amounts of unstructured data. The model brought advances in performance over previous neural networks architectures due to avoiding the need to employ dense matrix multiplications. The advantage gained in performance was that a single machine was able to train on over 100 billion words in a single day.

Further work in 2013 Mikolov et al. proposed further extension to the Skip-gram model with the Continuous Bag-of-Words Model (CBoW) and the Continuous Skip-gram Model.

The CBoW model uses a log-linear classifier with several historical and future words in the input with the goal of training the model to classify the current word in the criterion.

CBoW is defined as:

$$Q = N \times D + D \times \log2(V)$$

Where N = number of previous words, D = word representations, V = size of the vocabulary.

Continuous Skip-gram model attempts to classify a word based upon another word that exists in the same sentence. With the Continuous Skip-gram model, words are predicted before and after the current word in the vocabulary. Mikolov et al. found that the computational cost/complexity increased in predicting words with a greater range/distance from the current word and therefore applied a lesser weight to those words and sampled to a lesser degree based upon the assumption that words with a greater distance from the current word would usually have a lesser relation to the current word.

Continuous Skip-gram Model is defined as:

$$Q = C \times (D + D \times \log2(V))$$

Where C = max distance of the words, D = word representations, V = size of vocabulary.

Mikolov et al. go on to discuss that CBoW predicts the current word based upon context whereas the Continuous Skip-gram model predicts surrounding words given the current word as shown in Figure 15. (Mikolov, et al., 2013)

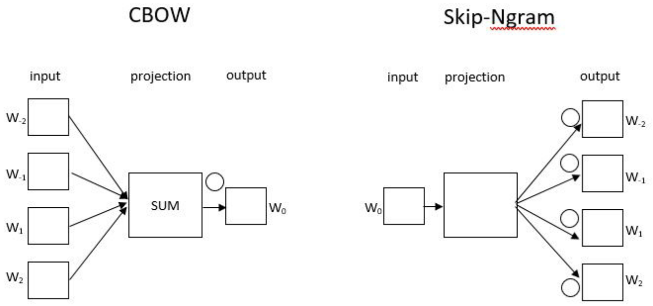


Figure 15: Continuous Bag of Words – Continuous Skip-gram model. Reproduced from Mikolov, et al. (2013)

The CBOW works best for words of a high frequency within a text whereas the Continuous Skip-gram model works better with words of a lower frequency ie: words that are rare within the text.

So, to predict a missing word in a text the CBoW model would be best suited to the task ie:

Text = “A man went to the bank to apply for a mortgage on his house”

If we were to omit the word “mortgage” from the text above, so that it read:

Text = “A man went to the bank to apply for a ‘--------' on his house”

Here the CBoW model would be able to predict the missing word of “mortgage” given the relations of the words around the missing word “mortgage”.

In the Continuous Skip-gram model we would pass the model the word "mortgage" and depending on the training of the model the related words would be returned with "bank" and "house" possibly being most prominent depending upon their relations and frequency within the training text. (Aggarwal, 2018)

Word2Vec is a shallow neural network model that can utilise both CBoW and Continuous Skip-gram models and can be trained on billions of words. Word2Vec uses a single hidden layer that can learn vector representations for words with similar distributional properties.

As in One-Hot Encoding, Word2Vec processes text by vectorising words (conversion of text to a numerical from) so that machines or indeed, neural networks can understand numerical representations better. The goal of Word2Vec is to group vectors of similar words/text together into a vector space. Word2Vec can detect similarities between vector spaces mathematically. Given this mathematical approach to detection of similarity between vector spaces it is possible to classify texts based upon their similarity or distance. (Tensorflow, 2021)

This measure of distance between texts or documents can be calculated using Cosine distance discussed later in this research.

GloVe is another learning algorithm that is used in the generation of word embeddings in a corpus. According to Pennington, et al., (2014) the algorithm works by "aggregating a global word-word co-occurrence statistics from a corpus and the resulting representations showcase interesting linear substructures of the word vector space". Unlike Word2Vec which operates and relies on local statistics GloVe can gather global statistics to acquire word vectors. The downside here is that the processing of a large corpus may be computationally expensive generating word embeddings for the whole corpus. Pre-trained embedding files are available for GloVe which are of various sizes and can be incorporated based upon the size of the corpus being worked upon. The available download files for GloVe embeddings are:

- Wikipedia 2014 + Gigaword 5 (6B tokens, 400K vocab, uncased, 50d, 100d, 200d, & 300d vectors, 822 MB download): glove.6B.zip
- Common Crawl (42B tokens, 1.9M vocab, uncased, 300d vectors, 1.75 GB download): glove.42B.300d.zip
- Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download): glove.840B.300d.zip
- Twitter (2B tweets, 27B tokens, 1.2M vocab, uncased, 25d, 50d, 100d, & 200d vectors, 1.42 GB download): glove.twitter.27B.zip

(Pennington, et al., 2014)

GloVe file glove.6B.100d.txt has been incorporated into the training of the neural networks built for sentiment analysis conducted in this research.

Term Frequency – Inverse Document Frequency (TF-IDF) provides a statistical measure that determines the relevance of a word in a single document or collection of documents. Initially TF-IDF was developed for document search and information retrieval. This works on the importance of a word within a document offset by the total documents containing the word.

Common words such as *"the", "this", "why", "what", "how"* for instance would rank very lowly in TF-IDF. Such common words would not add any semantic value to the document having a high frequency count across all documents. However, search terms such as "sex" may have a low frequency in a document but, a count across documents related to online grooming detection, such a word in a document would relate closely to the type of topic in the search criterion.

As previously discussed, there could be good cause to pre-process data before calculating the TF-IDF scores for a document, therefore the removal of common words known as “stopwords” which add little semantic value could prove to be beneficial here.

The measure of Term Frequency (TF) measures the frequency of a term in a document. There is the case that documents are not regular in length and therefore the frequency of a term may appear at a higher rate in documents of a longer length than those of a shorter length. Therefore, the calculation of term frequency is divided by the number of terms in the document. This measure provides normalisation across documents of variable length. (Spärck Jones, 1972)

TF is therefore calculated as: TF(i) = (frequency of term i in a document) / (total number of terms in the document j)

$$TF(i,j) = \frac{Term\ i\ Frequency\ in\ document}{\text{Total words in document}}$$

To calculate the IDF (inverse document frequency) score for a term i in a document j calculation of a higher weighting is applied to the term depending upon how rare the term is across all documents. Calculating the IDF score for a term is performed thus:

$$IDF(i) = \log_2\left(\frac{\text{Total documents}}{\text{documents with term i}}\right)$$

Combining the two calculations to calculate the TF-IDF for a term i across all documents containing term i would be thus:

$$\mathrm{W}_{\mathrm{i,j}} = \mathrm{tf}_{\mathrm{i,j}} * log\left(\frac{N}{df_i}\right)$$

$\mathrm{tf}_{\mathrm{ij}}$ = number of occurrences of i in j
dfi = number of documents where i exists
N = total number of documents
$\mathrm{W}_{\mathrm{i,}}$ = tf-idf score

(tfidf.com, 2021)

Using TF-IDF can be useful in calculating document similarity using the cosine measure of similarity by creating a vector of the words in a document and their relative TF-IDf scores. Considering the two sentences (documents) in Figure 16 below, we can calculate the TF-IDF score and create vectors based upon the scores.

Doc0: 'the car is driven on a motorway'
Doc1: 'the motorbike is driven on a road'

Figure 16: Sample sentences for TF-IDF

Using Python’s Sklearn library we can calculate the term frequency of each word in the documents and plot using a Pandas Dataframe. In Table 8 below the common words in each document carry a count of 1 in the row for each document. This, as previously discussed, highlights common or frequent words in the documents and therefore have little semantic value.

Table 8: TF-IDF, Term Frequency

	motorway	driven	motorbike	the	on	is	road	car	a
0	1	1	0	1	1	1	0	1	1
1	0	1	1	1	1	1	1	0	1

Using the term frequency counts we can then calculate the term frequency scores for each term in the two documents, these are displayed in Table 9.

Table 9: TF-IDF, Term Frequency Scores

	a	car	driven	is	motorbike	motorway	on	road	the
0	0.14	0.14	0.14	0.14	0.00	0.14	0.14	0.00	0.14
1	0.14	0.00	0.14	0.14	0.14	0.00	0.14	0.14	0.14

Again, using Python's SKlearn library we can go on to calculate the TF-IDF score. Note that in Table 10 the scores attributed to the common words in each document have now been attributed a score of zero and therefore have little importance in the document.

Table 10: TF-IDF Scores

	motorway	driven	motorbike	the	on	is	road	car	a
0	0.04	0.0	0.00	0.0	0.0	0.0	0.00	0.04	0.0
1	0.00	0.0	0.04	0.0	0.0	0.0	0.04	0.00	0.0

Furthermore, we can utilise the TfidfVectorizer in the Python SKLearn library to create a more machine-readable vector of TF-IDF and thus enable a more mathematical approach to be taken to calculating similarity. Column 1 in Figure 17 denotes the document and the word position.

```
(0, 0)        0.50
(0, 1)        0.35
(0, 2)        0.35
(0, 4)        0.50
(0, 5)        0.35
(0, 7)        0.35
(1, 1)        0.35
(1, 2)        0.35
(1, 3)        0.50
(1, 5)        0.35
(1, 6)        0.50
(1, 7)        0.35
```

Figure 17: TfidfVectorizer example.

TF-IDF can provide some important informative metrics about corpus data and has been used widely in information retrieval and document search. This approach can be adapted to identify key terms in a corpus that can then go on to form the basis for calculating document similarity metrics.

The notion of Artificial Neural Networks (ANN) is nothing new and can be said to have its origins in the early 1940s when McCulloch and Pitts wrote their article named "A logical calculus of the ideas immanent in nervous activity" (McColloch & Pitts, 1943). McCulloch and Pitts presented a mathematical model of neurons as basic switching elements in the human brain. The article aided in laying some of the foundations for artificial intelligence and neural networks (Anderson & Rosendfield, 1988).

Alan Turing, further, to pointing out the limits of intelligent machines in 1937 went on to define the Turing Test in 1950, which set out to establish whether a machine is truly intelligent. Turing also discussed genetic algorithms and learning machines in the same paper. (Turing, 1950). The first neural network was designed and built by Marvin Minsky in 1951. Minsky's neural network machine sported 3000 vacuum tubes and was able to simulate 40 neurons. (Ertel, 2017)

In humans, neural networks are networks of nerve cells in the brain, which amount to around 100 billion nerve cells. The human brain has the cognitive ability to learn a variety of capabilities such as motor and intellectual skills. According to Reber (2010) the human brain has a storage capacity of around 2.5 petabytes which in the average brain would amount to 2.5 million gigabytes of digital memory. So, emulating and recreating the cognitive abilities of the human brain can present several computational challenges not only in processing power required to recreate cognate functions of the brain but also the storage and processing large amounts of data.

In the figures below, two models of neural networks are shown, Figure 18 displays a biological model of a neural network whilst Figure 19 displays a formal model of the neurons within a network and the directed connections that exist between the neurons.

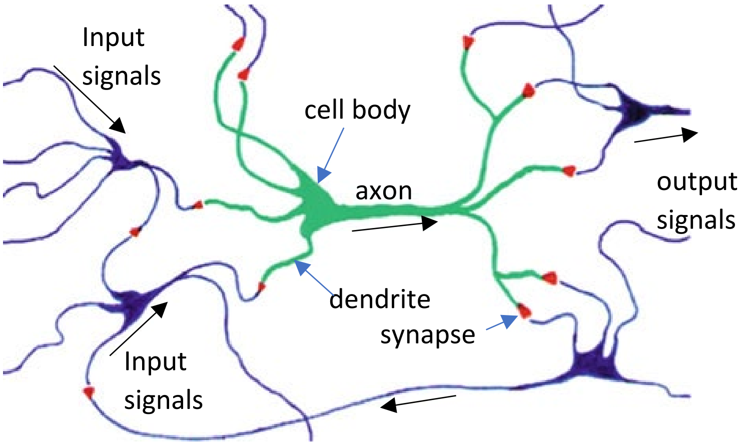


Figure 18: Biological model of a neural network. Adapted from (Ertel, 2017)

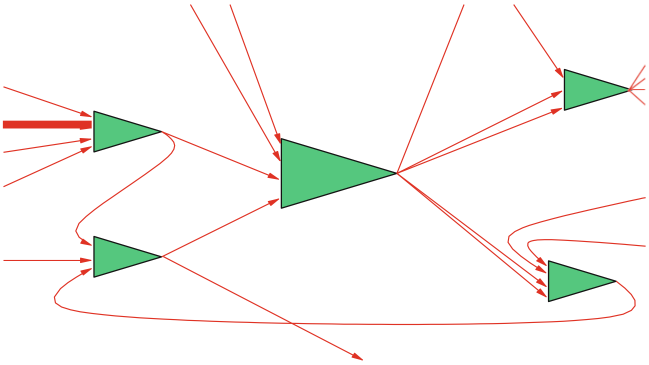


Figure 19: Formal model of a neural network. Reproduced from (Ertel, 2017)

The cell body of a neuron can store electrical charges much in the way that electronic devices can in capacitors or batteries. The electrical store of the cell body is loaded with electrical impulses from other neurons in the network. The higher the number of electrical impulses received from the other neurons build up a higher voltage in the cell body and thus (when the voltage exceeds its threshold) the neuron then fires. This firing is the unloading of the electrical charge and transmission of a charge over the axon and the synapses of the network to other neurons where the process is then repeated (Ertel, 2017). The synaptic connections between neurons are often strengthened in response to some form of external stimuli and such changes are the basis of cognition in living organisms (Aggarwal, 2018).

It becomes apparent that there is the opportunity for the simultaneous firing of neurons in the brain and therefore displays a high degree of parallel processing of information. This level of concurrency and randomness in the processing of information would also suggest that the brain's neural network operates in an asynchronous fashion. Given this level of parallelisation, it becomes apparent why Graphical Processing Units (GPU) are an ideal hardware solution for ANNs.

In a digital sense, neural networks exist as single and multi-layer neural networks. The single layer neural networks known as perceptron are learning algorithms geared towards binary classification where the perceptron can decide whether an input belongs to one class or another. Single-layer perceptron can be used in supervised learning tasks using linear classification where the primary goal is to predict which classification or category a particular data may belong to, based upon the input variables.

A typical plot for linear classification would fit with Figure 20 below, where there are two classes of data with a linear decision boundary used for classification of the input.

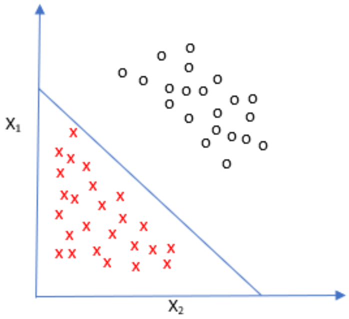


Figure 20: Linear Decision Boundary

Rosenblatt (1957) developed the perceptron model which considered the approach of flexible weight values applied to the neurons which proved useful in machines with adaptive capabilities. In the schema below the perceptron receives several inputs and weights which are then passed along to the Net input function. The function of the Net input function is to sum the multiplication of the inputs and weights and then pass the result to the activation function. In this penultimate stage the activation function will produce two different outputs for the model based upon the threshold set. The goal here is to emulate the function of neurons in the brain i.e.: "fire", or "don't fire". Figure 21 shows the schema for a 4-input perceptron.

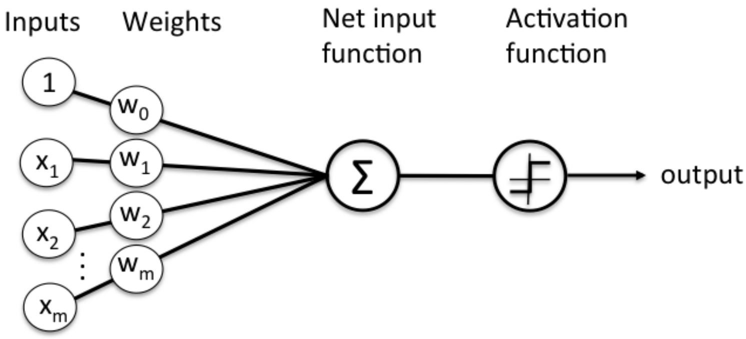


Figure 21: Perceptron schema. Reproduced from (Rosenblatt, 1957)

Typical linear activation functions used in perceptron are Linear and Step displayed in Figure 22 and Figure 23 respectively.

Linear:

Linear activation function

$$f(x)=ax$$

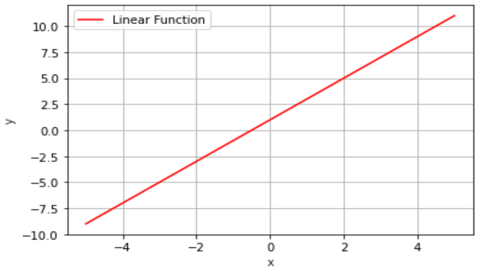


Figure 22: Linear function and graph

Step:

Step activation function

$$f(x) = \begin{cases} 0 \; for \;\; x < 0 \\ 1 \; for \;\; x \geq 0 \end{cases}$$

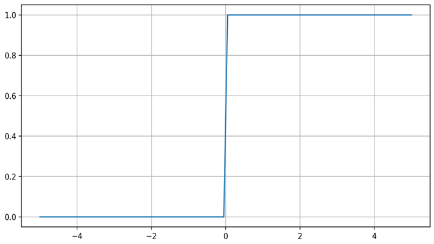


Figure 23:Step function and graph

Not all data in real world scenarios is linear and therefore a nonlinear approach to the processing/analysis of data needs to be taken. Nonlinear activation functions are generally used in the nonlinear transformation of data. This process ensures that representation within the given input space is mapped to a separate output space. Nonlinear activation functions are listed below.

Sigmoid:

Sigmoid function displayed in Figure 24 can be employed to transform a continuous space into a binary. Therefore, the function can take some real-valued data; a number for instance, represented here by (x) and uses a squash function to squash the data into a range of between 0 and 1. Large numbers (negative and positive) therefore have a proximity to 0. The goal here is to allow the final outputs of the final layer's activations to be interpreted as probabilities thus providing some notion of confidence in classification or prediction.

$$f(x) = \frac{1}{1+e^{-x}}$$

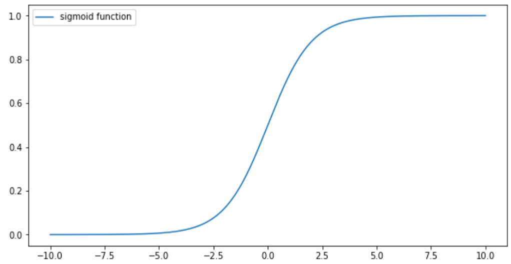


Figure 24: Sigmoid function and graph

Tan Hyperbolic (Tanh):

Like the Sigmoid function the Tan Hyperbolic (Tanh) displayed in Figure 25 is sigmoidal in shape and has a range of -1 to 1 therefore squashing the input (x) into the range given the value of (x). The advantage of using the Tanh function is that we can map strong negative inputs and zero inputs mapped near to zero in the Tanh graph as seen below. The tanh function is usually employed in the classification between classes.

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

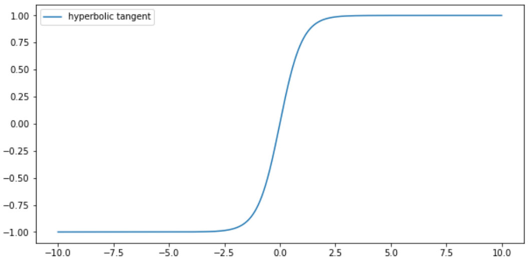


Figure 25: Tanh function and graph

Rectified Linear Unit (ReLU):

ReLU displayed in Figure 26 is a widely used activation function in the domain of Convolutional Neural Networks (CNN) / Deep Learning. Again, taking the input (x), the function converts (x) to 0 if (x) is a negative number. ReLU generally realises improved computation times over sigmoid and Tanh. However, one downside of using ReLU is that conversion of negative values to 0 can have adverse effects by not mapping negative values on the graph. To overcome this issue the Leaky me this issue the Leaky ReLU activation function can be employed within neural network models.

$$f(x)\begin{cases} 0\ for\ x < 0 \\ x\ for\ x \geq 0 \end{cases}$$

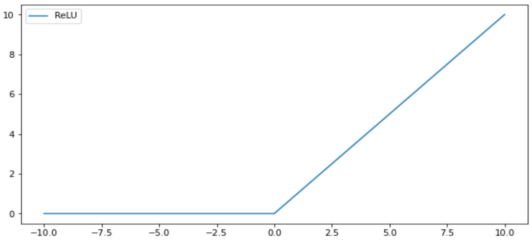


Figure 26: ReLU function and graph

Multi-layer perceptron (MLP), commonly referred to as Artificial Neural Networks, can, as the name suggests, utilise several what are known as hidden layers which can usually include two or more layers in the network. ANN were developed to overcome the apparent functional limitations of the single layer perceptron. Unlike the single layer perceptron MLP use a nonlinear activation function to classify data that may not be linearly separable.

In a typical MLP each node in a layer connects to every node in the next layer and so forth. It is due in part to the highly interconnected nature of multi-layered ANN that such feed-forward models can have efficient generalisation capabilities. ANN have proved to be very popular approaches in the classification of images (computer vision) and in NLP.

The number of layers (hidden layers) in an ANN can vary depending upon the type of operation the ANN is working upon. However, each layer will contain the same number of nodes which can be set or tuned to gain best performance from the ANN as can be seen in Figure 27. The output layer of the ANN can have K nodes depending upon the number of classes/classifications.

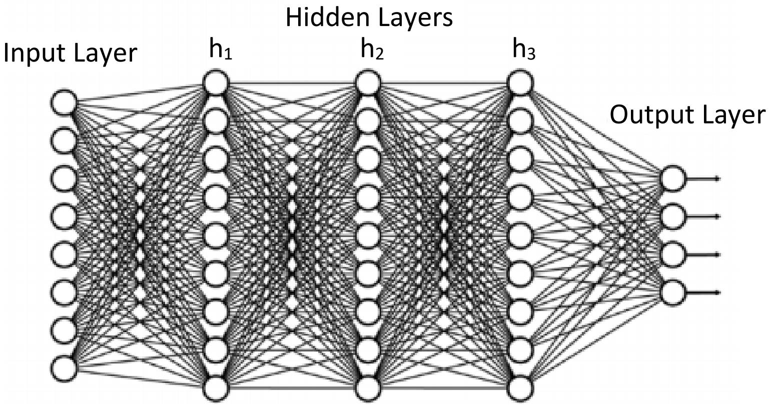


Figure 27: Multi Layer Perceptron. Reproduced from (Beysolow II, 2018)

Training of ANN usually involves presenting the model with a large amount of data to train and test on. Many scholars in the field of machine and deep learning have found that the performance of ANN increases with the amount of data presented whereas machine learning algorithms can return diminishing returns as the algorithm sees nothing new in the data. Bilgehan (2011) discussed the benefits that backpropagation included in the training of ANN and further development of neural network models that would include such an approach. Lample et al. (2016) also discuss the benefits to the training of ANN using back propagation in models focused on Named Entity Recognition (NER) in NLP showing that this has continued to be a popular approach.

Other aspects in the training of MLP models are:

- **Learning rates** – a static floating-point number that controls how weights are adjusted within a neural network when considering a loss gradient. Using a low learning rate may ensure that we “don’t miss anything”, however, this may mean that learn times are longer as the model takes a longer time in which to converge. Such rates are often manually adjusted, often using intuition or background knowledge of the domain application / previous experience of the data being worked upon. Smith (2017) argued that there could be cause to estimate a learning rate initially (very low) and then look to increase the rate with each iteration linearly or exponentially depending on the metrics returned as depicted in Figure 28.

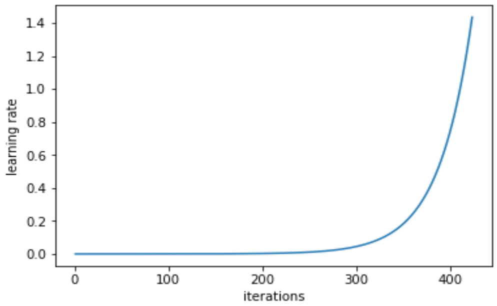


Figure 28: Learning Rate Graph. Reproduced from (Smith, 2017)

- **Epochs** – a hyper parameter that defines the number of iterations an algorithm will perform over an entire dataset. Epochs are traditionally large, ranging from 10 to 1000+. The goal here is to run the learning algorithm enough times thus allowing the number of errors from the model to have been minimized sufficiently.

- **Loss Function** – calculation of the number of times the model was incorrect. There are two mathematical approaches used in the calculation of Loss, which are Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). Each mathematically defined in Figure 29:

$$MSE = \frac{1}{N}\sum_{i-1}^{N}(h_\theta(x_i) - y^i)^2$$

$$RMSE = \sqrt{\frac{1}{N}\sum(h_\theta(x_i) - y^i)^2}$$

Figure 29: MSE and RMSE. Reproduced from (Beysolow II, 2018)

- **Training and Test data** – are subsets of the data used to:
    - Train a model, in the case of neural networks the calculation of weights and biases where a model will see and learn from the data presented.
    - Test and evaluate a final model fit onto the training set of data.
    - Indicative split ratio between train and test data may be 70 to 30 splits but this may need to be tuned given the performance metrics returned.

There are a variety of artificial neural network that exist.

- Feedforward Neural Network
- Radial basis function Neural Network
- Kohonen Self Organizing Neural Network
- Recurrent Neural Network (RNN)
- Deep Convolutional Neural Network (CNN)
- Modular Neural Network
- Generative Adversarial Network (GAN)

For this research, CNN and RNN will be discussed further and then tested for suitability in chapter 5.

Convolutional Neural Network is a deep learning architecture that can be applied in the fields of Computer Vision (CV) and NLP. Like other types of ANN, the CNN includes a number of layers in its construction. The layers of the network are constructed as laid out in Figure 30 below.

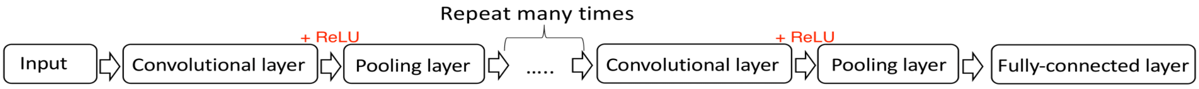


Figure 30: Basic Structure of a CNN. Reproduced from (Beysolow II, 2018)

The CNN is made up of Convolutional Layers and Downsampling/Pooling layers. Convolutional layers include a number of neurons that are able to scan their inputs for emerging patterns. The Downsampling/Pooling layers as seen in figure 30 are placed after the convolutional layers which aids in reducing the feature map dimensionality of the network which in turn improves computational efficiency and thus look to improve the performance of the network. In a typical CNN the two layers can appear in an alternate order, however, depending upon the application of the CNN this may not always prove to be the case. The Convolutional Layers and Downsampling/Pooling layers are then followed by a fully connected layer MLP.

As a simplified breakdown of CNN they can break down into three core concepts:

- Local receptive fields
- Shared weights and bias
- Activation and pooling

CNN, as previously mentioned can be applied to NLP in application areas such as sentiment analysis. With NLP, each word being presented to the CNN becomes an input for the CNN and is represented by a vector of a particular size – a vector size of 7 as shown in Figure 31 sentence matrix.

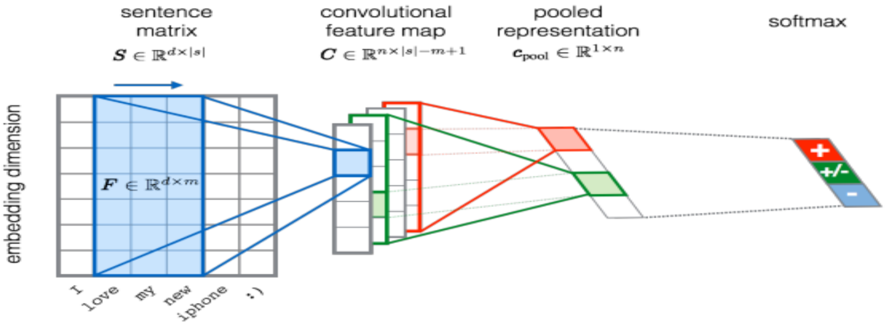

sentence matrix
$S \in \mathbb{R}^{d\times|s|}$
convolutional feature map
$C \in \mathbb{R}^{n\times|s|-m+1}$
pooled representation
$c_{pool} \in \mathbb{R}^{1\times n}$
softmax
embedding dimension
$F \in \mathbb{R}^{d\times m}$
I
love
my
new
iphone
:)
+
+/-
-

Figure 31: CNN used in NLP Sentiment Analysis. Reproduced from (Yin et al. 2017)

Each word or token in the sentence is placed into the matrix as a low-dimensional representation generated as a vector by utilising word2vec or GloVe. The convolutional layer of the network is used for representation learning from sliding w-grams. (Yin, et al., 2017)

Yin et al. (2017) compared a range of Deep Neural Network Architectures which measured the performance of CCN in various NLP applications (sentiment analysis/classification, text classification tasks) against other architectures such as RNN and LSTM. The results of the research carried out by Yin, et al. found that CNN and RNN performance was dependent on the importance of understanding the semantics of a whole sequence and that whilst the learning rate changes CNN performs at a smooth rate. However, changes to the hidden size and the batch size in CNN could result in large fluctuations in performance. The research conducted by Yin et al. could prove valuable if a CNN were to be employed in the classification of predatory discourse.

Recurrent Neural Networks (RNN) were developed to overcome the issues presented by Feed-Forward Neural Networks (FFNN). The issues related to FFNNs were related to the following:

- Poor handling of sequential data.
- Focused upon the current input and considers nothing else.
- Unable to memorise values from previous inputs.

RNN can process sequential data whilst accepting current data and data from previous inputs. RNN include an internal memory that enables them to store and thus remember previous inputs. Unlike FFNN, RNN share parameters across the layers of the network having the same weighting within each layer of the network which differs to FFNN where each node carries a different weighting. The weightings, however, can still be adjusted using backpropagation and gradient descent to incorporate reinforcement learning.

Another characteristic of RNN is Back Propagation Through Time (BPTT) where RNNs can calculate errors from the output layer to the input layer and therefore adjust parameters of the model as required.

Two known issues with RNN are Exploding Gradients and Vanishing Gradient which are defined by the size of the gradient relating to the slope of loss along an error curve usually when the gradient is too small. When the gradient is too large the issue of Exploding Gradient occurs and results in an unstable model where weightings become too large and thus end up represented as NaN.

RNN can be classified as four distinct types:

- One to One – generally used for ML problems and includes a single input and single output in its construction.
- One to Many – used for problems where multiple outputs can be derived from a single input.
- Many to One – takes in a sequence of information and produces a single output such as may be required in sentiment analysis.
- Many to Many – takes in sequences of information as its inputs and then outputs the data as a sequence. This type of RNN is typically used in Machine Translation where text sequences in one language are converted to another ie: English to French.

Long Short-Term Memory (LSTM) is a variant of RNN and is capable of learning long-term dependencies in sequences of data. LSTM is able to process the whole sequence of data and is a popular approach used in NLP for sentiment analysis, speech recognition, and machine translation to name but a few applications.

LSTM was first introduced by Hochreiter & Schmidhuber (1997) and was a novel approach to addressing the error backflow problems encountered in Backpropagation Through Time (BPTT) and Real-Time Recurrent Learning (RTRL). LSTM approach to maintaining long-term dependencies is to introduce a memory cell into the network. As with RNN the LSTM model employs a range of repeated modules for each time step. Furthermore, the output module of each time step is controlled by a set of gates as a function of the old hidden state.

LSTM as discussed by Lui & Guo (2019) consists of an input gate (i) that controls the size of new memory contents. A forget gate (f) that makes decisions on the quantity of data that may need to be forgotten. The model has an output gate (o) that modulates the quantity of memory output and finally a cell (c) avtivation vector which includes two component parts – forgotten previous memory and a modulated new memory . A simple schematic of an LSTM model is displayed in Figure 32.

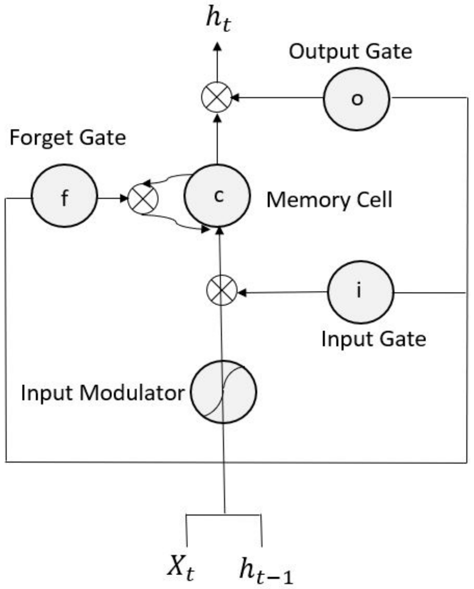


Figure 32: Simple LSTM schematic with weight matrices (arrows). Reproduced from (Lui & Guo, 2019)

Bidirectional LSTM or BiLSTM as commonly known is a sequence processing model and employs two LSTM models in its design. In BiLSTM, one LSTM takes the sequence forwards, and the second LSTM moves the sequence in reverse through the model which allows BiLSTM to access preceding and succeeding contexts in the sequence. The gain here is the quantity of data available to the network which therefore improves the context available to the model. This allows the model to know the words that immediately follow or precede the current word in the sentence the model is working upon. Lui & Guo, (2019) discuss that LSTM and BiLSTM are popular approaches in text classification with BiLSTM displaying greater performance over LSTM.

Figure 33 below is comparative illustration of the LSTM and BiLSTM models.

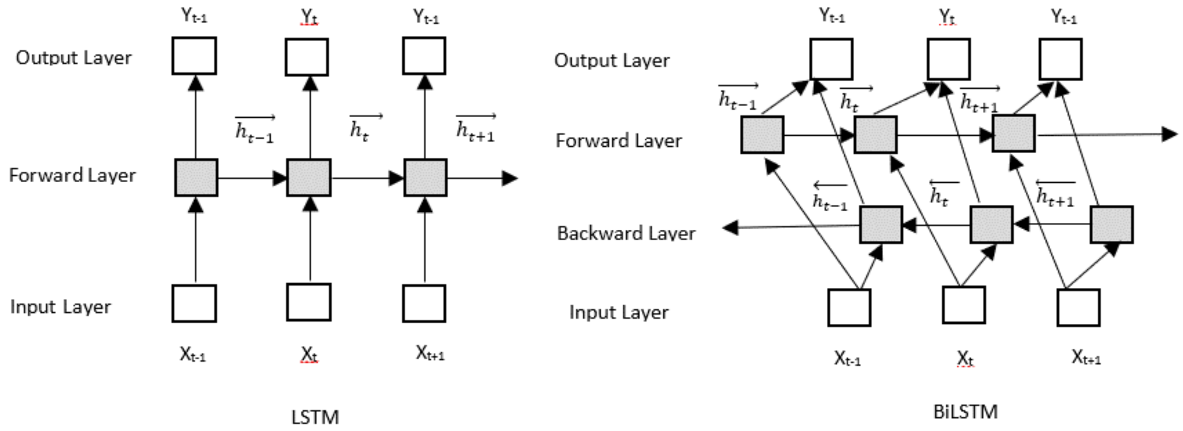


Figure 33: LSTM and BiLSTM models. Reproduced from (Lui & Guo, 2019)

4 NLP Frameworks, Tools and Algorithms

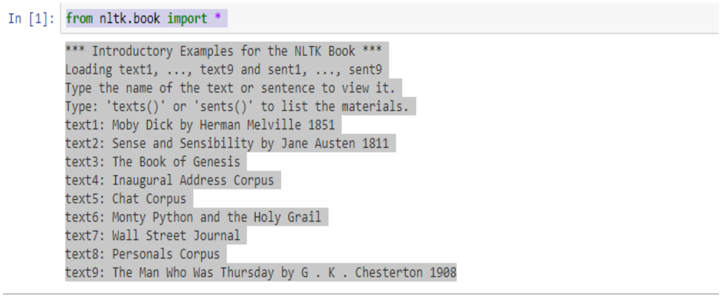
In this section we will discuss some of the frameworks, tools, and algorithms available that enable the development of solutions for data pre-processing, tagging, stemming and lemmatization, classification, and sentiment analysis.

NLTK:

The Natural Language Tool Kit (NLTK) is a fully developed tool kit with a rich array of features that enable the analysis, modelling and interpretation of digital discourse using a suitable Python programming environment.

Using NLTK makes it possible to store and process data allowing a corpus to be worked on in full or in part dependent upon the initial results received or the manner of the task being carried out.

In Figure 34 below, is an example of how NLTK is able to import a corpus and display sample data from the corpus.



```
In [1]: from nltk.book import *

*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

Figure 34: NLTK Corpus Import. Reproduced from (NLTK, 2021)

NLTK includes processing libraries for tokenization, stopword removal, classification, stemming, part of speech (POS) tagging, concordance, collocation, dispersion, lexical diversity, and is a popular inclusion in many NLP solutions. (NLTK, 2021)

Textblob:

Textblob is a simplified text-processing library for Python which is built on NLTK and Pattern. Textblob offers a simple API that enables the simple deployment of a range of NLP tasks as detailed below.

TextBlob features:

- Noun phrase extraction
- Part-of-speech tagging
- Sentiment analysis
- Classification (Naive Bayes, Decision Tree)
- Tokenization (splitting text into words and sentences)
- Word and phrase frequencies
- Parsing
- n-grams
- Word inflection (pluralization and singularization) and lemmatization
- Spelling correction
- Add new models or languages through extensions
- WordNet integration

TextBlob (2021)

Using powerful Python commands from the TextBlob it is possible to analyse a piece of text or extract interesting features. Such easy-to-use TextBlob commands can enable quick solutions to NLP problems or data analysis tasks as shown in the example below using Naïve Bayes classifier for sentiment analysis of a text.

```
from textblob import TextBlob

review = TextBlob("Textblob is a really amazing Python library that makes NLP fun !")

print (review.sentiment)
```

Result:

```
Sentiment(polarity=0.49, subjectivity=0.55)
```

Tensorflow:

Tensorflow, created by the Google Brain Team is an open-source library that facilitates the building of machine learning and deep learning models. TensorFlow can process data converted into multidimensional arrays of higher dimensions known as Tensors. The employment of multidimensional arrays allows for the handling of high volumes of data.

Tensorflow supports a range of languages including Python and JavaScript and includes processing via Central Processing Unit (CPU) or Graphical Processing Unit (GPU) via Nvidia Cuda Cores.

(Tensorflow, 2020)

Keras:

Keras is a deep learning API written in Python that runs on top of TensorFlow. The aim of Keras was to enable researchers and developers to perform experiments in a shorter timeframe. Keras offers a high-level API of TensorFlow and according to Keras (2021) offers four key capabilities:

- Execute Tensor operations on CPU, GPU, and TPU.
- Compute gradient of arbitrary differentiable expressions.
- Provides scalable computation across many devices.
- Export of programs to servers, browsers, mobile and embedded devices.

(Keras, 2021)

There are several algorithms that can be employed in the field of data science and NLP for the purpose of calculating similarity measure, classification, and sentiment analysis. The focus of this research are the algorithms employed in sentiment analysis, and text similarity measures to establish whether a document can be classified as a grooming conversation or not. Two similarity measures have been focused below.

Cosine Distance:

Cosine distance is the metric used to calculate the similarity between documents. In a mathematical sense it is the cosine measure of the angle between two vectors that exist in a multi-dimensional space. In simple terms the cosine similarity measure is to determine whether two vectors existing in the same space point in a similar direction.

Documents, as previously discussed, can be represented as vectors (such as the vectors created using TF_IDF vectorizer) and therefore the numeric similarity between the vectors can be calculated.

However, there is the issue of term frequency vectors suffering from becoming long in length and suffering from sparsity (containing many zeros in the vector). With Cosine Similarity it proves advantageous in calculating the similarity of documents based on the words that exist in both documents and thus ignore zero matches. It becomes apparent here that pre-processing techniques such as stemming, and lemmatisation become invaluable in ensuring that the document vectors carry the same vectors of words that have been stemmed to their root/base form. (Han, et al., 2012)

The mathematical formula for Cosine Similarity is shown below:

$$Cosine\ Similarity = \frac{\sum_{i=1}^{n} x_i\, y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\ \sqrt{\sum_{i=1}^{n} y_i^2}}$$

A typical graphical representation of Cosine similarity can be seen below in Figure 35.

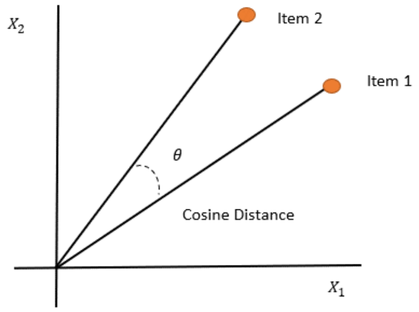


Figure 35: Graphical Representation of Cosine Similarity/Distance. Reproduced from (Damgeti, 2017)

Support Vector Machines (SVM):

SVM is a supervised learning algorithm that can be used in classification tasks in NLP. With SVM it is possible to plot data items as points in a n-dimensional space where n represents the number of features resident within the data which hold the value of a coordinate within the n-dimensional space. Classification is calculated by finding a hyper-plane differentiating between two classes as shown in Figure 36.

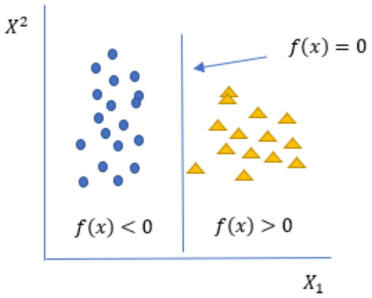


Figure 36: SVM two classes with hyper-plane. Reproduced from (Al Amrani, et al., 2018)

SVM suffer from performance issues when dealing with large data sets where training times can be considerably high. However, SVM performs well where there is a clear separation between classes and proves to work well in situations where the number of dimensions in the data is larger than the qty of samples available.

Al Amrani et al. (2018) tested the performance of SVM in sentiment anlysis where they achieved an 81% accuracy in detecting sentiment in 1000 reviews. Al Amrani, et al. improved this measure through their use of a Random Forrest Support Vector Machine (RFSVM) achieving 83.4% accuracy in detecting sentiment in 1000 reviews.

Padurariu & Breaban (2019) investigated the imbalance of text classification and found in cases of linear classification SVM performed well when employing Bag Of Words (BoW) as part of the process of classification. Futher results found that using GloVe and Word2Vec embeddings did not perform well when there was a reduced size in data and shorter length of text in the corpus used. This provides valuable insight when considering the grooming data used later in this research as the length of a typical document in the grooming corpus is much smaller when compared to that of the other corpora obtained for analysis. In consideration of low sample numbers within texts it may prove valuable to consider the work of Kou et al. (2020) who highlighted some of the issues associated with small sample sizes and high dimensionality.

Naïve Bayes:

Sang-Bum et al. (2006) discuss that Naïve Bayes had proved to be a popular approach in machine learning for a number of years. Research using Naïve Bayes in text classification over the same period where documents had been considered as binary feature vector identifying whether a particular word had been present or not and was known as the Multivariate Bernoulli Naïve Bayes. However, a more popular approach had been to use a Multinomial model due to the issue of the standard traditional model not being able to utilize term frequencies in documents. In calculating the probability of a text for text classification tasks the traditional Bayes’ theorem is applied as seen in Figure 37.

$$p(c|d_j) = \frac{p(d_j|c)p(c)}{p(d_j)} = \frac{p(d_j|c)p(c)}{p(d_j|c)p(c) + p(d_j|\overline{c})p(\overline{c})} = \frac{\frac{p(d_j|c)}{p(d_j|\overline{c})}\cdot p(c)}{\frac{p(d_j|c)}{p(d_j|c)}\cdot p(c)+p(\overline{c})}$$

Figure 37: Bayes' Theorem. Reproduced from (Sang-Bum, et al., 2006)

Sang-Bum et al. (2006) also discuss that Multinomial Naïve Bayes can suffer from poor performance where the qty of training examples is low or where documents have a long length and as part of the process all documents are merged into one larger document which then serves as a unique example for the model. Therefore, the model can perform better with shorter documents where a fewer number of terms participate in the estimation/classification. Given the document lengths of the grooming data used in this research, investigation and testing Multinomial Naïve Bayes may prove a worthwhile endeavour.

4.1 Summary

The technology for analysing digital discourse and the huge amounts of data it creates presents opportunity to apply machine learning and deep learning architectures in the detection or classification of online grooming discourse.

Whilst frameworks and tools offer platforms to conduct sentiment classification, document classification, and document similarity measures, there lacks a dedicated platform directly focused on the detection of malicious interlocution in online discourse. Rich data sets, prelabelled with sentiment or other classifications enable the development of mature solutions in a variety of applications in NLP.

However, as the research has shown, there is a distinct lack of available data relating to online grooming discourse which limits the opportunity to compare/benchmark the grooming data acquired against other grooming corpora. Whilst other corpora are available for analysis, the apparent issue is the character length of each interaction and the topics discussed within the data with a degree of disparity in the analysis metrics returned later in this research. Although, the latter could prove valuable in providing a novel solution to the problem with both grooming and general chat corpora which further provides opportunity to test the effectiveness of any detection strategies employed in a mixed chat environment.

Furthermore, the variance in the data available requires the data to be normalised in such a way that any intelligent solution has a uniform view of the data. The impact of such normalisation could affect the context of the data being presented or in total data loss within a document depending on the length of the document presented for normalisation.

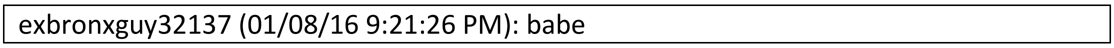
At present, there is no “sure fire application” protecting children across a range of platforms by analysing discourse transparently and raising parental alerts . The experiments used in this research look to inform an approach that would realise the construction of a more novel solution to the problem.

5 Malicious Interlocutor Detection Using Forensic Analysis of Historic Data.

5.1 Data Sources

As part of this research data was acquired from four of the previously mentioned chat corpora sources. In each instance the data was parsed from its original format the extract the raw chat data and then imported into the RDBMS (MariaDB) for further processing/pre-processing and analysis.

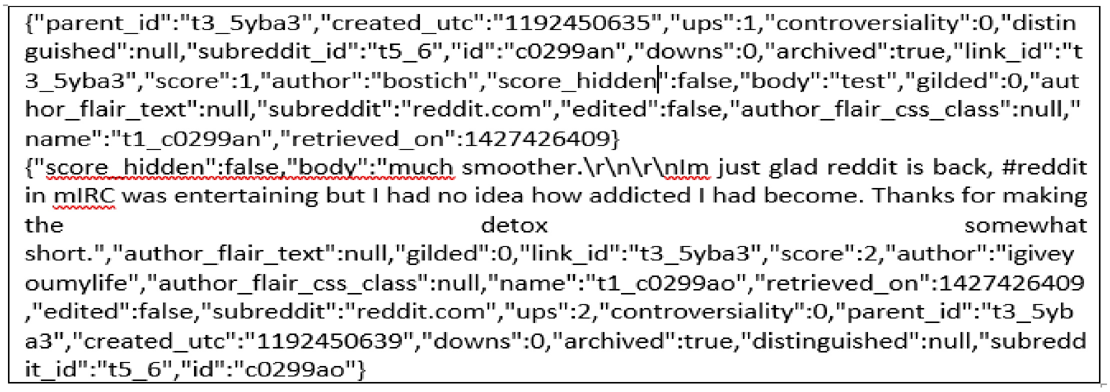
Perverted-Justice – 291 case files were extracted from the *Perverted-Justice* data centre which equated to 380,012 records. The data is available as text files in the data centre which can then be parsed to extract the raw chat data as required. Permission to access the *Perverted-Justice* data centre and credentials can be seen in appendix 1. An example of the raw data is displayed in Figure 38 below:



```
exbronxguy32137 (01/08/16 9:21:26 PM): babe
```

Figure 38: Sample raw data Perverted Justice Corpus

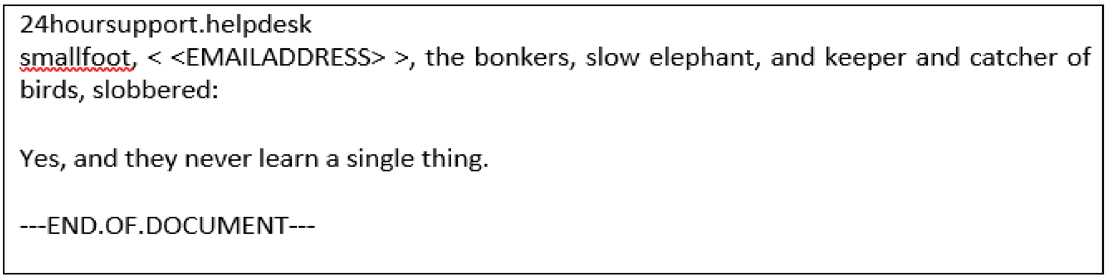
Reddit – 1.2TB of data was acquired and parsed resulting in 491m records. An example of the raw Reddit data is displayed in figure 39 below:



```
{"parent_id":"t3_5yba3","created_utc":"1192450635","ups":1,"controversiality":0,"distin
guished":null,"subreddit_id":"t5_6","id":"c0299an","downs":0,"archived":true,"link_id":"t
3_5yba3","score":1,"author":"bostich","score_hidden":false,"body":"test","gilded":0,"aut
hor_flair_text":null,"subreddit":"reddit.com","edited":false,"author_flair_css_class":null,"
name":"t1_c0299an","retrieved_on":1427426409}
{"score_hidden":false,"body":"much smoother.\r\n\r\nIm just glad reddit is back, #reddit
in mIRC was entertaining but I had no idea how addicted I had become. Thanks for making
the                              detox                              somewhat
short.","author_flair_text":null,"gilded":0,"link_id":"t3_5yba3","score":2,"author":"igivey
oumylife","author_flair_css_class":null,"name":"t1_c0299ao","retrieved_on":1427426409
,"edited":false,"subreddit":"reddit.com","ups":2,"controversiality":0,"parent_id":"t3_5yb
a3","created_utc":"1192450639","downs":0,"archived":true,"distinguished":null,"subredd
it_id":"t5_6","id":"c0299ao"}
```

Figure 39: Sample raw data Reddit Corpus

Westbury Labs Corpus – approximately 38gb of data was acquired and parsed resulting in 170m records. An example of the raw data is displayed in Figure 40 below:



```
24hoursupport.helpdesk
smallfoot, < <EMAILADDRESS> >, the bonkers, slow elephant, and keeper and catcher of
birds, slobbered:

Yes, and they never learn a single thing.

---END.OF.DOCUMENT---
```

Figure 40: Sample raw data Westbury Labs Corpus

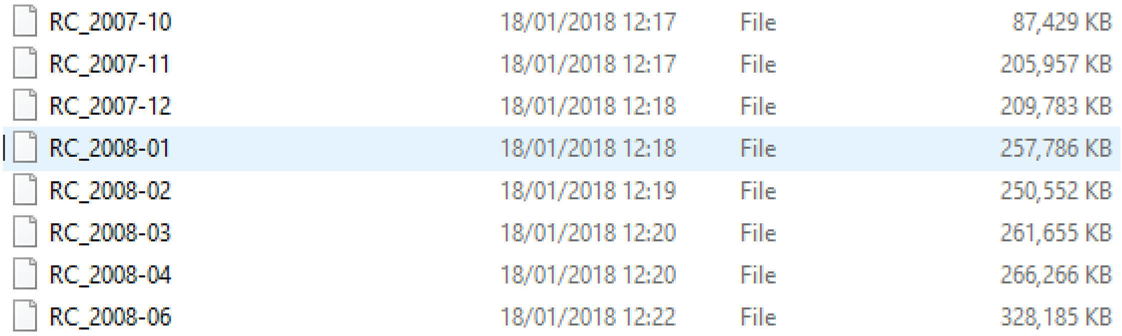
Twitter Archive – 631gb of twitter data was acquired but after initial assessments of the data the data files were abandoned due to the multilingual nature of the discourse within the files as can be seen below. Dealing with multilingual data is currently outside of the scope of this research.

"Dana ya estas hablando con la tatuadora para hacerte algo nuevo? \n\nS"

Further, after parsing from its JSON format there was a great reduction in size due to the amount of unnecessary data within the files. In the September 2016 archive which equated to 426gb of data the final size of the collated files after parsing was 372.96mb.

5.1.1 Parsing and processing the raw data:

Collection of the Westbury and Reddit corpora required the combination of separate files into one larger file for processing. This larger file could then be parsed to extract the raw chat data and write out to a text file for later import into the MariaDB database. A sample data file list for the Reddit corpus can be seen Figure 41 below.



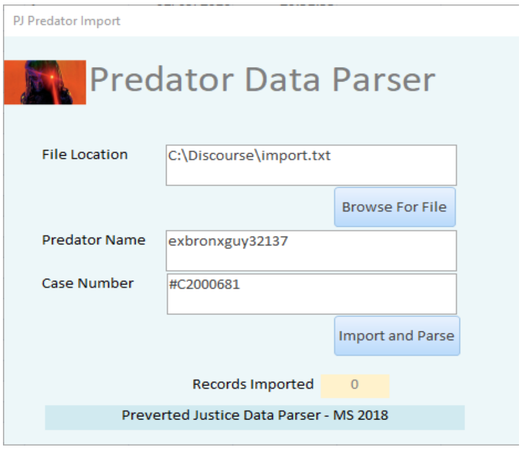
RC_2007-10	18/01/2018 12:17	File	87,429 KB
RC_2007-11	18/01/2018 12:17	File	205,957 KB
RC_2007-12	18/01/2018 12:18	File	209,783 KB
RC_2008-01	18/01/2018 12:18	File	257,786 KB
RC_2008-02	18/01/2018 12:19	File	250,552 KB
RC_2008-03	18/01/2018 12:20	File	261,655 KB
RC_2008-04	18/01/2018 12:20	File	266,266 KB
RC_2008-06	18/01/2018 12:22	File	328,185 KB

Figure 41: Sample Reddit datafiles for extract

To extract the chat text from the files Java routines were written to isolate the chat text and write out the isolated data for import to MariaDB later. A sample of the Java processor for *Twitter* data can be seen in Appendix 3.

With the *Preverted-Justice* data the decision was taken to extract individual files rather than web scrape the Perverted-Justice web site archive. With downloading individual case files from the datacentre, it was possible to identify the Predator and Victim participants and ensure a uniform extract of the data. The Perverted-Justice datacentre contained case files of the chat room (meet me), mobile phone text messages, voice messages, and sexually explicit images sent from the Predator to the Victim. For this research, just the chat room discourse files were imported and processed.

To process the Perverted-Justice data files a Microsoft Access application was written (PJ Predator Import) to import the data directly into the MariaDB database via a MySQL ODBC connector and is shown in Figure 42.



PJ Predator Import

Predator Data Parser

File Location: C:\Discourse\import.txt

Predator Name: exbronxguy32137

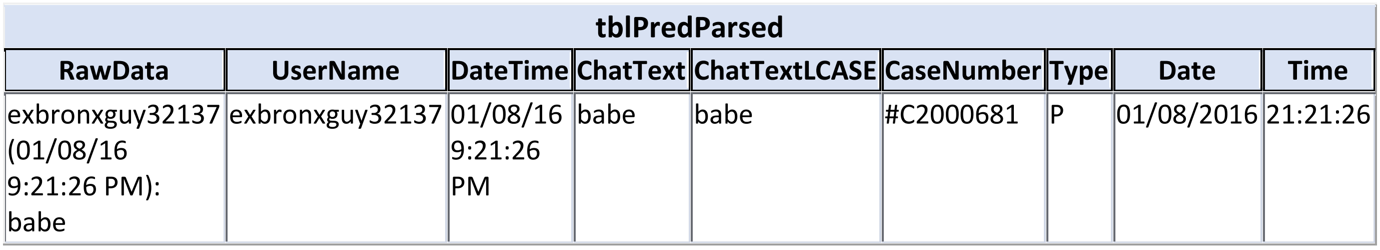
Case Number: #C2000681

Records Imported: 0

Preverted Justice Data Parser - MS 2018

Figure 42: Predator Data Parser

All data from the raw text was preserved and parsed out into its component parts as seen in Figure 43 below.



tblPredParsed								
RawData	UserName	DateTime	ChatText	ChatTextLCASE	CaseNumber	Type	Date	Time
exbronxguy32137 (01/08/16 9:21:26 PM): babe	exbronxguy32137	01/08/16 9:21:26 PM	babe	babe	#C2000681	P	01/08/2016	21:21:26

Figure 43: Sample predator data import.

During the import process, to aid with future sentiment analysis of the data the application made best attempt to replace any text emoticons with their text equivalences using the list of emoticons that had been compiled in Appendix 2 along with lower casing the text before import to the database. From the Predator username identified on the import form the application was able to label the data as a type "P" for predator and "V" for victim to assist with anonymising reference to the participants in further processing and analysis of the data.

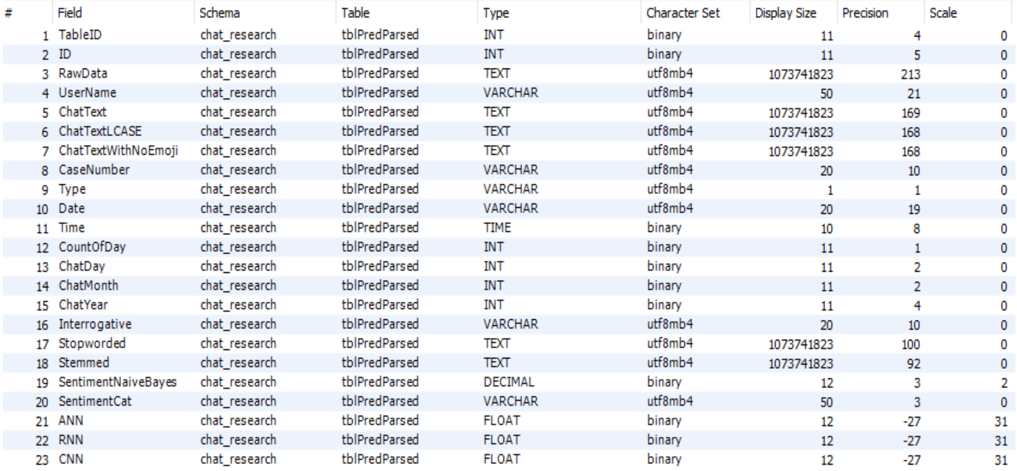
Other processes such as establishing the length of time of the grooming processes using a count of the days during the import was discarded as there were variable lengths of the groom and later establishing that perpetrators of online grooming are a heterogenous group and operated in different time frames it was deemed that analysing such data had little significance.

5.2 Experimental Testbed

To investigate opportunities for prevention of malicious conversations the following testbed was used to conduct the experiments.

Hardware Resources

- Server Specification: DELL R410 (x3) – 16GB, 2 x XEON 5650 2.66ghz - 8 cores per CPU, 2TB HDD , 1GB Dual Nic
- MariaDB Database Server – ucbccluster.blackburn.ac.uk. Data table design below. UTF8 character encoding used across data files and within the data tables for consistency and to avoid any anomalies on import. Figure 44 displays the design of the Predator table in MariaDB.



#	Field	Schema	Table	Type	Character Set	Display Size	Precision	Scale
1	TableID	chat_research	tblPredParsed	INT	binary	11	4	0
2	ID	chat_research	tblPredParsed	INT	binary	11	5	0
3	RawData	chat_research	tblPredParsed	TEXT	utf8mb4	1073741823	213	0
4	UserName	chat_research	tblPredParsed	VARCHAR	utf8mb4	50	21	0
5	ChatText	chat_research	tblPredParsed	TEXT	utf8mb4	1073741823	169	0
6	ChatTextLCASE	chat_research	tblPredParsed	TEXT	utf8mb4	1073741823	168	0
7	ChatTextWithNoEmoji	chat_research	tblPredParsed	TEXT	utf8mb4	1073741823	168	0
8	CaseNumber	chat_research	tblPredParsed	VARCHAR	utf8mb4	20	10	0
9	Type	chat_research	tblPredParsed	VARCHAR	utf8mb4	1	1	0
10	Date	chat_research	tblPredParsed	VARCHAR	utf8mb4	20	19	0
11	Time	chat_research	tblPredParsed	TIME	binary	10	8	0
12	CountOfDay	chat_research	tblPredParsed	INT	binary	11	1	0
13	ChatDay	chat_research	tblPredParsed	INT	binary	11	2	0
14	ChatMonth	chat_research	tblPredParsed	INT	binary	11	2	0
15	ChatYear	chat_research	tblPredParsed	INT	binary	11	4	0
16	Interrogative	chat_research	tblPredParsed	VARCHAR	utf8mb4	20	10	0
17	Stopworded	chat_research	tblPredParsed	TEXT	utf8mb4	1073741823	100	0
18	Stemmed	chat_research	tblPredParsed	TEXT	utf8mb4	1073741823	92	0
19	SentimentNaiveBayes	chat_research	tblPredParsed	DECIMAL	binary	12	3	2
20	SentimentCat	chat_research	tblPredParsed	VARCHAR	utf8mb4	50	3	0
21	ANN	chat_research	tblPredParsed	FLOAT	binary	12	-27	31
22	RNN	chat_research	tblPredParsed	FLOAT	binary	12	-27	31
23	CNN	chat_research	tblPredParsed	FLOAT	binary	12	-27	31

Figure 44: Predator Data Table design

- Galera High Availability Database Cluster with HA-Proxy Load Balancer. The figure below outlines the network structure for the database cluster in the Cluster Laboratory at Blackburn University Centre. The HA Proxy node employs a round-robin load balancer which proved adequate for a single user instance. Galera cluster instantaneously replicates data across all nodes in a Master-to-Master configuration. Gigabit ethernet used as the backbone for the interconnection of all nodes. Figure 45 displays the network designed for the Galera cluster used in the research.

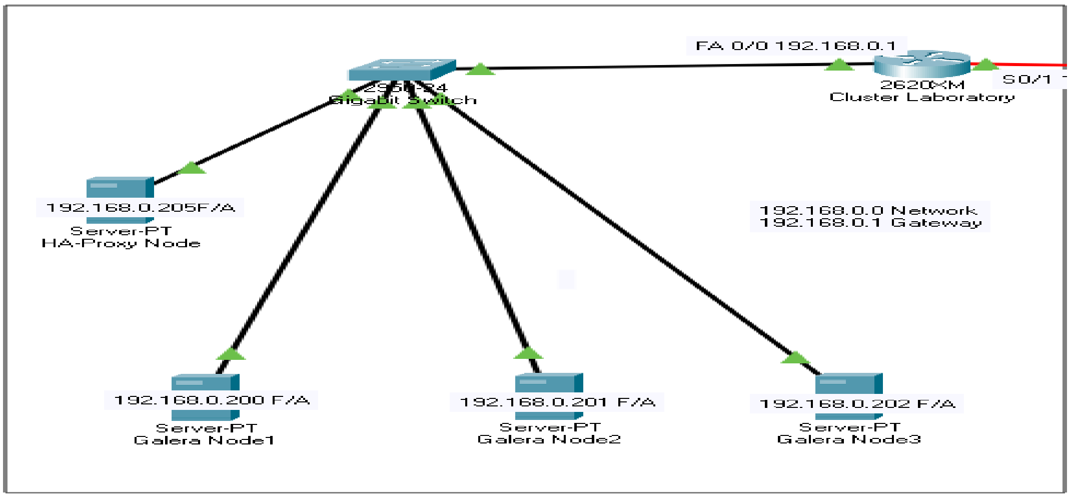


Figure 45: Galera Cluster network diagram

- Client-Side Hardware specification – AMD 8350 Bulldozer 4.1ghz, 16GB RAM, 240GB SSD, 2 x 2TB HDD, 1GB Nic

Software tools used in the research

- Anaconda: Spyder, Jupyter Notebook for Python development of analysis and detection tools.
- #LancsBox: Analysis of parsed data, Word and Feature Counts, Collocations, Bigram and Trigram analysis.
- MS Access – Import and parsing of Perverted-Justice grooming data files.
- Java – Import and parsing of *Twitter*, *Reddit* and *Westbury Labs* corpus files.

5.3 Data Pre-Processing performance

Using the pre-processing workflow previously outlined the grooming data was updated within the database populating additional data fields so the effects of pre-processing could be measured. Two fields were populated Stopworded and Stemmed (Porter Stemmer) using the Python NLTK library. As found with other operations performed on the database, the updating of data on a row-by-row basis temporally costly.

The data was extracted from the database as a csv file and worked on externally to the database. The csv file was later imported to the database and the Stopworded and Stemmed fields updated in the predator data table by performing an equijoin based on the row ID field.

Given the short length of the chat discourse which averaged 21 chars across the grooming data there was a loss in data thus creating 14506 null rows. These null rows would be ignored in any future extracts to capitalise on the metrics that could be extrapolated.

There was significant loss in the number of tokens post stopword removal with a total 695299 tokens removed. The positive metric round in processing via csv file is that the processing of the 376760 rows took just over 50 seconds which is vast improvement over the number days it could potentially take iterating through the data table in the database.

Tables 11 and 12 outline the performance metrics of pre-processing performing stopwording only, and stopwording and stemming of the data. The only effect on the data here was the increase in processing time which is to be expected given the additional operations performed.

Table 11: Effects of Pre-Processing on the data: STOPWORDS (NLTK)

Rows Processed:	376760
Text Rows Output after stopwording (removal null rows):	362254
Total Rows lost after stopwording:	14506
Total words/tokens imported in the file:	1683901
Total words/tokens exported to file:	988602
Total words/tokens lost after stopwording:	695299
Execution Time in seconds:	50.24

Table 12: Effects of Pre-Processing on the data: STOPWORDS (NLTK) and STEMMING – Porter Stemmer

Rows Processed:	376760
Text Rows Output after Stopwording/Stemming (removal null rows):	362254
Total Rows lost after Stopwording/Stemming:	14506
Total words/tokens imported in the file:	1683901
Total words/tokens exported to file:	988602
Total words/tokens lost after Stopwording/Stemming:	695299
Execution Time in seconds:	80.39

Table 13 below highlights the top ten tokens of three files – the raw data, the data after being stopworded and, the data having been stopworded and stemmed. There is a distinct difference in the token lists after processing with the removal of common tokens such as “I” and “to” for instance.

Tokens such as “u”, “im”, “ur” could be added to a custom stopword list and removed, whilst tokens such as “lol” could be replaced by their text equivalent “laugh out loud” which could potentially have a positive effect on any sentiment analysis performed.

Table 13: Top Ten Tokens for each file: Raw Data, STOPWORDS (NLTK), STEMMED – Porter Stemmer

Token Raw Data	**Token Frequency**	**Token Stopworded (NLTK)**	**Token Frequency**	**Token Frequency**	**Token Stopworded/Stemmed**
i	81023	u	70025	u	70025
u	69809	lol	35254	lol	35260
to	35658	ok	17008	like	17612
lol	35237	like	16992	ok	17270
you	32670	im	13004	im	13031
me	24872	ur	11807	ur	12127
it	24073	want	10600	want	12025
a	21810	dont	10377	dont	10377
do	19368	get	8861	get	9951
that	18135	happy	8655	happi	8664

Table 14 compares the effects of stemming on the data using two stemmers Porter and Lancaster. The aim of this comparison was to glean some notion of the aggressiveness of each stemmer and their effect on the data. As can be seen in the top ten tokens listed there are only marginal differences.

Table 14: Top Ten Token comparison after Stopword and Stemming (Porter Vs Lancaster)

Token Stopworded/Stemmed Porter	Token Frequency	Token Stopworded/Stemmed Lancaster	Token Frequency
u	70025	u	70025
lol	35260	lol	35318
like	17612	lik	18202
ok	17270	ok	17302
im	13031	im	13330
ur	12127	ur	12261
want	12025	want	12040
dont	10377	ye	11655
get	9951	dont	10382
happi	8664	get	9973

Whilst pre-processing of textual data can have positive effects on the varied operations that can be performed such as sentiment and similarity, removal of high frequency tokens could potentially have negative effects in cases where the lengths of document in a corpus are low as with the grooming data acquired for this research. This research has already discussed that in some cases pre-processing techniques can have negligible positive effects on NLP operations. Approaches such as TF-IDF or Word Embeddings may prove to be more beneficial in such cases. Each of the latter approaches have been tested within the scope of this research.

5.4 Experiments

To investigate opportunities for prevention of malicious conversations the following experiments were conducted.

5.4.1 Experiment 1: Sentiment analysis of online grooming data in RDBMS

The aims of the following tests were to detect whether grooming conversation was positive or negative and whether this could aid in identifying grooming discourse. In the OGDM model previously discussed there is opportunity for negative discourse centred around parents in the Isolation phase of grooming where the predator looks to isolate the victim from the meaningful people in the victim's life. In addition to the testing of the OGDM model there is opportunity to test the performance of ANN, CNN, RNN and Naïve Bayes identified previously.

The ANN, CNN and RNN were trained on the IMDB review data set with 25,000 positive reviews and 25,000 negative reviews resident in the data. To maintain parity in the training of the three models the same training procedure was undertaken which included:

- pre-processing the review data (refer to pre-processing workflow).
- tokenization and padding to a max length.
- Import GloVe word embeddings file - glove.6B.100d.txt. to obtain vector representations.
- Train the models over 5 Epochs – the models generally began to overfit beyond the 5th epoch.

In each case two tests were conducted:

1) Sentiment analysis of 30,000 rows in the database and then calculate the mean score for the sample.
2) Sentiment analysis of a single case file from the grooming data and detect whether there was any change in the sentiment analysis where a larger number of tokens were presented in one analysis.

Two metrics have been returned which outline the performance of each model:

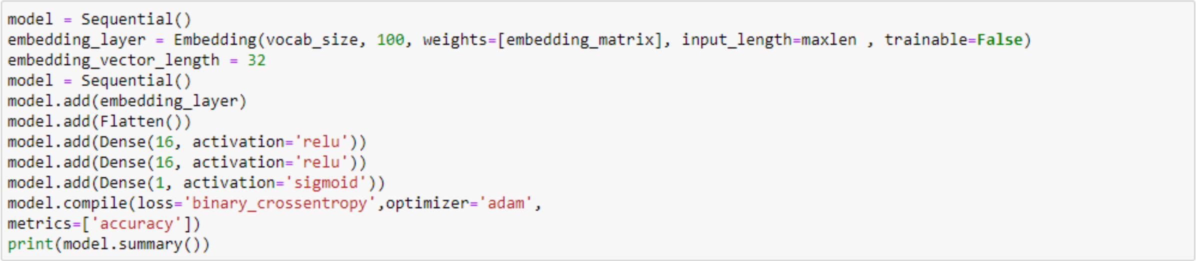
a) Loss: measure of error, calculated using RMSE as previously discussed.
b) Accuracy: which calculates the fraction of predictions made correctly by the models using the calculation below:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

As a comparison, the Naïve Bayes sentiment classifier from the Textblob Python library was also used to establish whether anything new could be gleaned from using a different approach to the sentiment analysis.

Test 1 - ANN:

The ANN model in Figure 46 was trained on IMDB review data.



```
model = Sequential()
embedding_layer = Embedding(vocab_size, 100, weights=[embedding_matrix], input_length=maxlen , trainable=False)
embedding_vector_length = 32
model = Sequential()
model.add(embedding_layer)
model.add(Flatten())
model.add(Dense(16, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy',optimizer='adam',
metrics=['accuracy'])
print(model.summary())
```

Figure 46: Test 1 ANN Model

Figure 47 displays the accuracy of the ANN model after performing 5 epochs.

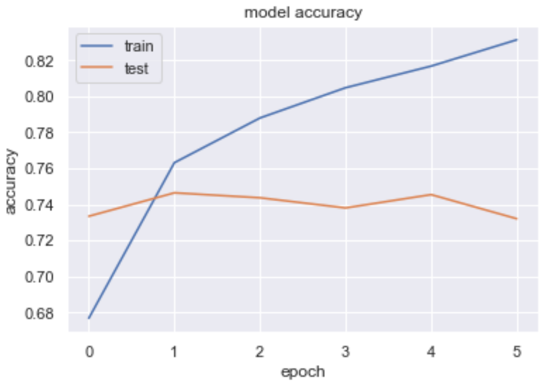


Figure 47: ANN Model Accuracy Graph

Figure 48 displays the loss of the ANN model after performing 5 epochs.

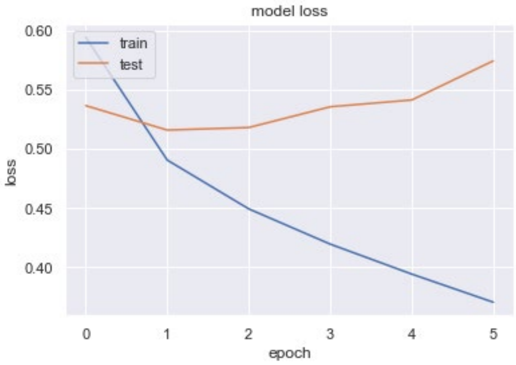


Figure 48: ANN Model Loss Graph

ANN training performance:

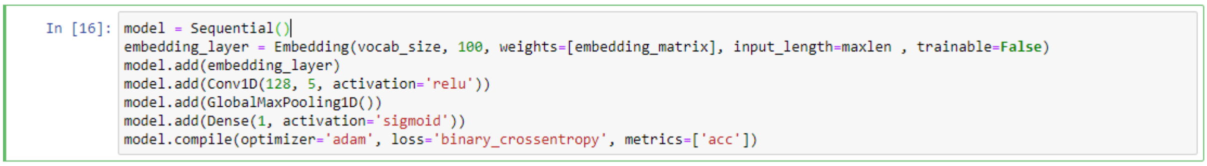
The model was tested on 30,000 rows in the database to detect the sentiment of the grooming discourse for both predator and victim. The results of the sentiment analysis using a simple ANN can be seen below. Iterating through the data to detect sentiment on a row-by-row basis proved temporally costly as previously discovered, with iterating through the grooming data table taking over a day to run. This proved to be the case across all three models and therefore an alternative method was required for tests on larger sets of the data. To overcome the impact of updating the data table row-by-row a copy of the data was written out to a csv file which showed a marked performance increase in time of 0.74 second speedup per row. Writing of the sentiment data to csv has been adopted for the testing of CNN and RNN. Table 15 displays the performance metrics for ANN.

Table 15: ANN Performance Results

Total Epochs	5
Model Accuracy	74.04%
Average Sentiment Score Single Row 30K sample	68%
Average Runtime per row in RDBMS	0.78 seconds
Sentiment Score for Single Case presented as whole text (Case Number – 622)	84%
Runtime for single case via text file import	0.18 seconds
Average Runtime per row writing to file	0.04 seconds

Test 2 CNN:

The CNN model shown in Figure 49 was trained on IMDB review data.



```
In [16]: model = Sequential()
         embedding_layer = Embedding(vocab_size, 100, weights=[embedding_matrix], input_length=maxlen , trainable=False)
         model.add(embedding_layer)
         model.add(Conv1D(128, 5, activation='relu'))
         model.add(GlobalMaxPooling1D())
         model.add(Dense(1, activation='sigmoid'))
         model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
```

Figure 49: CNN Model

Figures 50 and 51 display the accuracy and loss of the CNN model after performing 5 epochs.

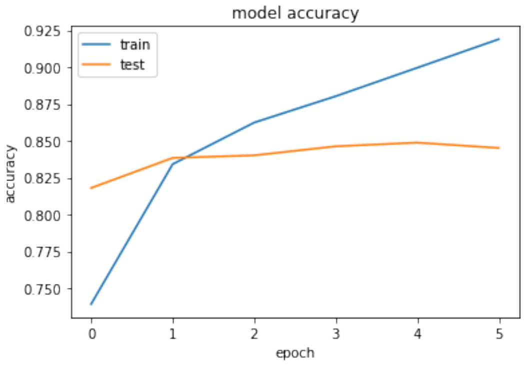


Figure 50: CNN Model Accuracy Graph

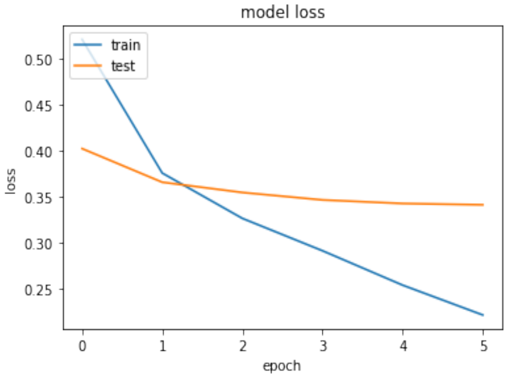


Figure 51: CNN Model Loss Graph

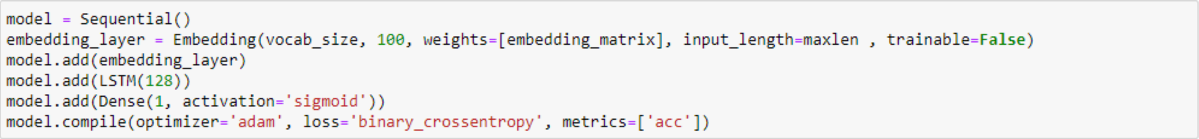
Results:

Table 16: CNN Performance Results

Total Epochs	5
Model Accuracy	85.3%
Average Sentiment Score Single Row 30K sample	82%
Average Runtime per row in RDBMS	0.84 seconds per row
Sentiment Score for Single Case **(Case Number – 622)**	99%
Average Runtime per row writing to file	0.053 seconds per row

Test 3 RNN(LSTM):

The CNN model shown in Figure 52 trained on IMDB review data.



```
model = Sequential()
embedding_layer = Embedding(vocab_size, 100, weights=[embedding_matrix], input_length=maxlen , trainable=False)
model.add(embedding_layer)
model.add(LSTM(128))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['acc'])
```

Figure 52: RNN(LSTM) model

Figures 53 and 54 display the accuracy and loss of the RNN model after performing 5 epochs.

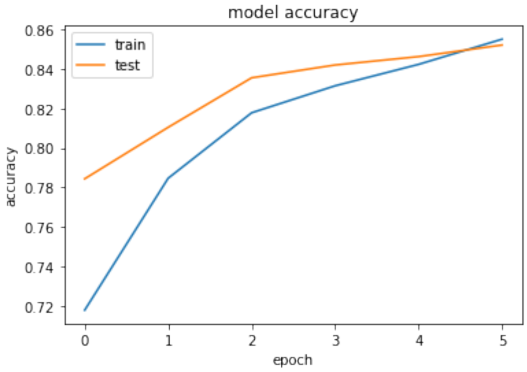


Figure 53: RNN(LSTM) Model Accuracy Graph

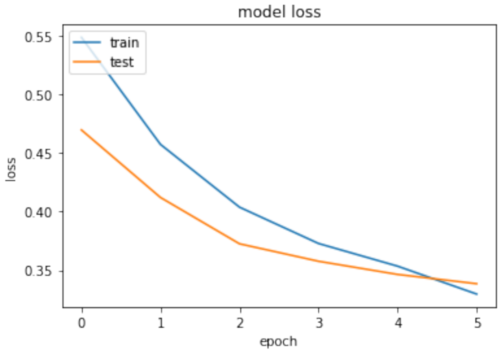


Figure 54: RNN(LSTM) Model Loss Graph

Results:

Table 17: RNN(LSTM) Performance Results

Total Epochs	5
Model Accuracy	93.4%
Average Sentiment Score Single Row 30K sample	76.37
Average Runtime per row in RDBMS	0.88 seconds per row
Sentiment Score for Single Case (CaseNumber – 622)	64% - positive
Average Runtime per row writing to file	0.06 econds per row

5.4.2 Experiment 2: TextBlob - Naïve Bayes Sentiment Classifier:

Using the Naïve Bayes classifier included in the TextBlob library, it was easier to extrapolate the polarity of the discourse as the classifier uses the following measure: < 0 = Negative, 0 = Neutral, >0 = Positive.

The Naïve Bayes classifier differed from the predominantly positive score of the neural network models, however, even though the results sit in the neutral range it still identifies that the discourse is not negative and is consistent with the early grooming models discussed earlier. The results of the Naïve Bayes classifier experiment can be seen in Table 18.

Results:

Table 18: Comparison of the sentiment of each participant.

Participant	Sentiment	Qty	Participant	Sentiment	Qty	Difference
P	Neg	1579	V	Neg	1462	7.41
P	Neu	180033	V	Neu	181069	-0.58
P	Pos	6242	V	Pos	6374	-2.11

The results shown in Table 19 show that testing the OGDM negative discourse directed towards parents during the isolation phase proved that in this corpus the discourse was again predominantly neutral to positive showing that the Perverted-Justice data did not directly follow the OGDM.

Results:

Table 19: Discourse directed towards parent (mom,mum,dad,father)

Sentiment	Qty Rows Relating to Parent	Percentage Sentiment Relating to Parent
Neg	63	1.14%
Neu	5337	96.53%
Pos	129	2.33%

5.4.3 Experiment 3: Interrogative Classification using Naïve Bayes in NLTK – Analysis.

The aim of this experiment was to glean some sense of the type of discourse taking place beyond that of being positive or negative. Early in the research, as found by Seedall, et al., (2019) the level of interrogation within the grooming data had a bias of the predator interrogating the victim based on a test set of 1800 lines. As can be seen in the analysis detailed in Table 20 and Figure 55 this trend remained consistent across the whole of the data with around 15% differential between Predator and Victim.

One of the most informative metrics to be returned from the analysis was the amount of emotive discourse used by the Victim compared to the Predator. In keeping with the sentiment analysis carried out, the bulk of the exchanges between the two participants were classified as “Statement” of which 149,587 rows fell into the neutral classification.

Table 20: Classification of discourse within grooming corpus (comparing Predator / Victim)

Participant	Type Discourse	Qty	Participant	Type Discourse	Qty	Difference Victim to Predator
P	Accept	5517	V	Accept	7498	+35.91%
P	Bye	1543	V	Bye	2632	+70.58%
P	Clarify	34327	V	Clarify	34243	-0.24%
P	Continuer	1232	V	Continuer	583	-52.68%
P	Emotion	6563	V	Emotion	13236	+101.68%
P	Emphasis	3064	V	Emphasis	3314	+8.16%
P	Greet	2006	V	Greet	2019	+0.65%
P	nAnswer	5895	V	nAnswer	6874	+16.61%
P	Other	840	V	Other	1172	+39.52%

P	Reject	5569	V	Reject	4295	-22.88%
P	Statement	79492	V	Statement	77174	-2.92%
P	System	1367	V	System	1373	+0.44%
P	whQuestion	19211	V	whQuestion	17058	-11.21%
P	yAnswer	4562	V	yAnswer	1479	-67.58%
P	ynQuestion	16666	V	ynQuestion	15955	-4.27%

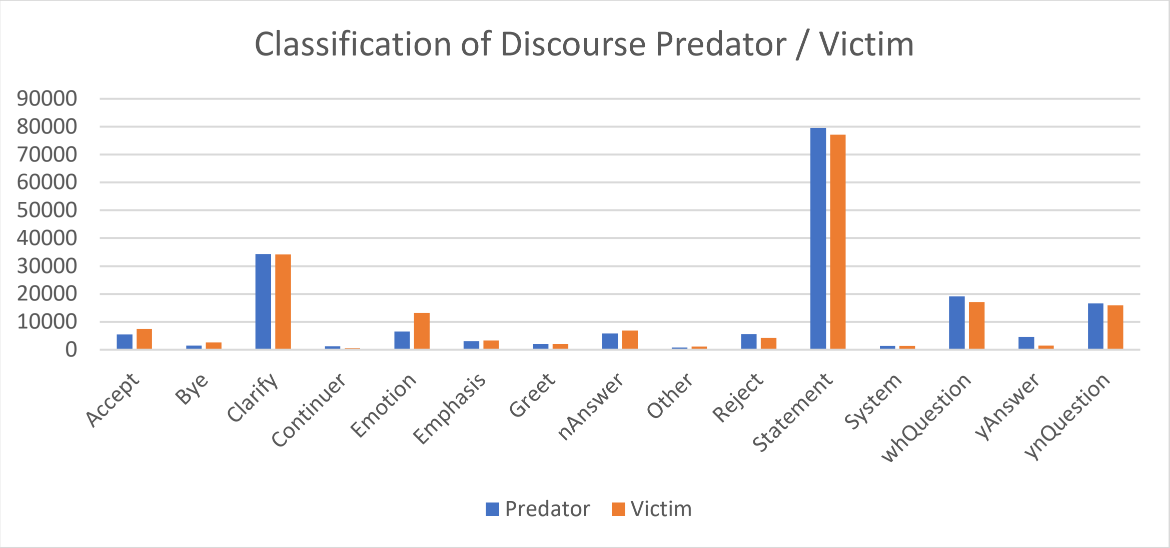


Figure 55: Graph of discourse type - grooming corpus

5.4.4 Experiment 4: Cosine, SVM, Naïve Bayes similarity for predator detection

Cosine Similarity:

To test the effectiveness of similarity measures in detection of grooming conversation the Cosine Similarity of three data files were compared. Using TF-IDF to determine the most informative features in each text a three-way comparison was conducted.

Three text files were used in the following tests.

1. Predator Victim Training Data – PredTrain.csv
2. Predator Victim Test Data – PredTest.csv
3. Westbury Corpus Data – Corpus.csv

Test 1: Perform Cosine Similarity with Pre-Processing of data files.

Each file was pre-processed using the following workflow:

- Remove csv formatting.
- Remove non alpha characters.
- Remove any single characters (tokens such as "u" as previously identified in the pre-processing performance).
- Remove multiple spaces.
- Remove Stopwords.
- Apply stemming with the Porter Stemmer.

Results:

Cosine Similarity Result: PredTest = 76.1% similarity

Corpus = 13% similarity

Test 2: Perform Cosine Similarity without Pre-Processing (csv formatting removal remained)

Results:

Cosine Similarity Result: PredTest = 76.8% similarity

Corpus = 5.7% similarity

SVM and Naïve Bayes:

A test set of data was created using Predator and Corpus data. Using a supervised learning approach, the data had labels assigned to each row to identify which corpus it originated from. The data was pre-processed, split between training and test data (70 to 30 split) and then vectorised using TF-IDF. The accuracy then calculated on the vectorised test data created. The results of the SVM Naïve Bayes test can be seen in Table 21.

Results:

Table 21: Results of SVM and Naive Bayes Similarity

Algorithm	Score
Naïve Bayes Multinomial	82.8%
SVM	84.3%

5.4.5 Observation and other analysis of the data

The words “babe” and “baby” are common terms of endearment used in the discourse and appear throughout the grooming corpus; however, this may not have been the case in other chat corpora. To test this, an indicative list of what could be deemed as prominent features in grooming data was analysed using dispersion plots. The list of tokens passed for analysis are words deemed conducive with grooming discourse fitting with the OGDM model previously discussed. The list focuses on three key areas:

a) Deceptive Trust Development / Isolation - terms referring to parent(s): “mom”,”dad”,”parents”.

b) Deceptive Trust Development - terms of endearment: “babe”, “baby”.

c) Sexual Gratification – terms relating to sex: “sex”.

The tokens were analysed in each corpora (*Perverted-Justice*, *Wesbury Labs*, and *Reddit*) and the results returned can be seen in Figures 56, 57 and 58 respectively. It is evident in the analysis returned that the tokens analysed had a greater prevalence throughout the grooming data than that displayed by the more general chat corpora and therefore suggests there is a noticeable difference in some of the context and terms used in the grooming discourse.

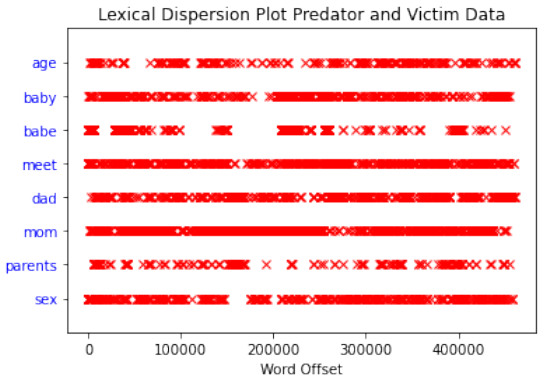


Figure 56: Lexical Dispersion Plot Perverted-Justice data

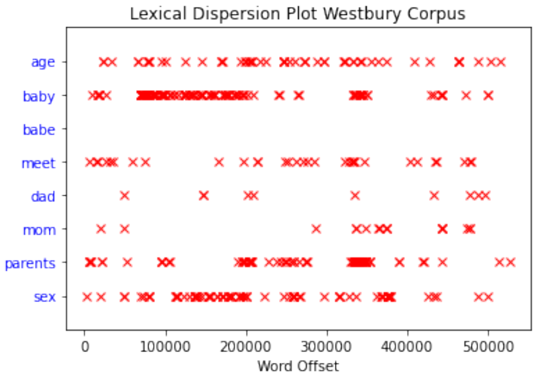


Figure 57:Lexical Dispersion Plot Westbury Corpus

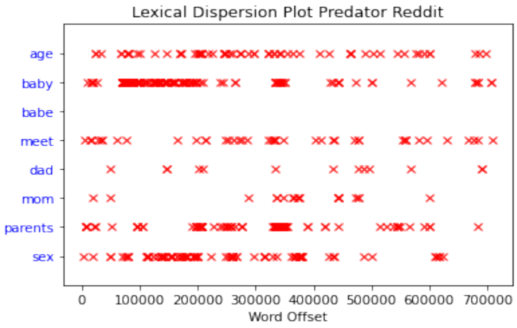


Figure 58: Lexical Dispersion Plot Reddit Corpus

In performing a feature count across all three corpus (*Perverted-Justice*, *Reddit*, *Westbury Labs*) which returned over 189,000 unique tokens there were 159 variants of the word “babe” in the feature counts returned, there were often concatenated messages such as “babeorbbackthought”, “babereasonwho”, “babewaynoolove”. Of the 159 variants 147 were unique to the *Perverted-Justice* corpus.

The feature counts conducted also revealed that there were 36661 unique tokens in the *Perverted-Justice* corpus, which from observation were made up of concatenated words. A list of variants for “babe” can be seen in Appendix 6.

It must be noted that words unique to grooming and possibly indicative of grooming stages can appear in both grooming and more general chat corpora.

5.4.6 Reflection on limitations of experiments conducted.

Reflection on sentiment analysis:

The accuracy of sentiment of the neural networks and naïve bayes model tested displayed varying results in accuracy. However, the sentiment analysis performed on each model is consistent in determining that the discourse used in online grooming is predominantly positive.

The testing of the OGDM and the sentiment directed towards parents by the Predator proved to be positive and was therefore inconclusive based on the dataset used in the analysis.

The use of word embeddings (GloVe) performed well and aided in returning positive training metrics used in the experiments. This could form part of the process in future development of the networks and their use in a more mature solution.

The use of prelabelled data (positive and negative) used in the training of the neural networks for sentiment analysis has shown that this approach enabled the required outcomes. Therefore, the potential labelling of grooming data with labels which highlighted whether documents in the discourse are predatory or not could prove to be an operation carried out in future work. Other research centred around the grooming data acquired has taken similar approaches and therefore a similar form of labelling could be a prospect.

Interrogative Classification using Naïve Bayes in NLTK – Analysis:

The experiment in the main was successful in providing insight on the type of discourse taking place, however, the aim of detecting whether there was a high degree of interrogative discourse taking place between Predator and Victim did not highlight that there was nothing out of the ordinary when considering how the Predator establishes initial contact and performs a reconnaissance to establish a safe environment to proceed.

An interesting metric to consider is that the Victim discourse had a 100% higher emotion classification than the Predator. Further analysis of this classification needs to take place to drill down into the type of emotion and what the emotion relates to. A point to consider here however, is that the Victim in this case is an adult (agent provocateur) posing as a child and therefore the discourse is not a true reflection of that of a child. This measure of emotion may be skewed by the fact this is an adult-to-adult exchange.

Cosine, SVM, Naïve Bayes similarity for predator detection:

This experiment showed some success in testing the effects of TF-IDF and the similarity between corpora. The experiments, although conducted on the corpora acquired for analysis, were not conducted on single documents within the corpora nor were they tested interacting with the database. But, in the methods applied there was a positive identification of a grooming conversation.

This is an issue that could arise if the texts were static texts and chat interactions have a very dynamic constantly shifting and evolving lexicon. As chat conversation evolves, such as the use of emoticons to convey feelings, emotions, gestures, and voice inflection, the data used for similarity measures would need to evolve also.

Observation and other analysis of the data:

In the analysis of potentially unique terms such as those highlighted in experiment 4 proved that the lexicon of grooming and general discourse is similar. Whilst the word “babe” proved that there were terms in the corpora that may be unique and help to identify grooming discourse, this could be said for other corpora where the chat platform has a particular focus such as technology or sport for instance.

6 Conclusion and Future Work

The overarching aim of this research was to investigate a variety of technological solutions that would be able address the research question of;

“Malicious Interlocutor Detection Using Forensic Analysis of Historic Data.”

The landscape of child protection online is one of parental controls and various legislation that places emphasis of responsibility on social media providers. On-going advertisement campaigns on UK television channels point parents to valuable resources for child protection and highlight the plight of children online and their vulnerabilities. The statistics reviewed in this research have highlighted to scale of the problem and that although new legislation is brought into existence as a means of dealing with perpetrators, the penalties and potential sentences doled out to those who are apprehended do not seem to act as an effective deterrent, even though new legislation looks to keep in line with current technologies and trends.

Given the current upward trends in online predation of children this research has found the current legislative measures questionable in their effectiveness and more needs to be done by social media platforms and other technology giants to ensure government guidelines and legislation are adhered to.

Through the collection and processing of available chat corpora from *Westbury Labs Corpus, Reddit,* and *Preverted-Justice* it has been possible to investigate NLP techniques and test the performance of pre-processing techniques carried out on textual data. Using such techniques, it was possible to test the validity of some of the published work relating to the effects of pre-processing and it is the findings of this research that in some cases the effects were indeed negligible. In some cases, there was negative impact on the data with the loss of document content due to the short character length of the text in the grooming corpus. Notably, this measure of loss was related to stopword removal applied in documents consisting of one to two words.

The effect of stemming on the data does reduce feature space by representing a base form of a word throughout the corpus therefore having a positive effect on word embeddings and the vectorisation techniques tested.

The *Perverted-Justice* corpus has enabled the research to analyse the sentiment of grooming discourse and establish that in the main such discourse is deemed to be positive and therefore does not follow the pattern of other discourse such as bullying or radicalisation where the discourse could be predominantly negative. However, the labelled data used to train neural networks in sentiment analysis has identified the opportunity for future work on the grooming data acquired where labels can be placed on the data to provide neural networks with a supervised learning approach to classifying discourse as predatory. This has been successful during the testing of similarity measure where documents from *Perverted-Justice* and *Westbury Labs* were labelled and using Cosine Similarity, SVM, and Multinomial Naïve Bayes the test returned successful results.

The use of neural networks in NLP classification tasks is, as the published works suggest, a tried and tested means of accurately classifying documents. Whilst there are variances between the training accuracies of the models tested, each model corroborated the findings of positive and negative sentiment classification of text data. The only limitation here, again, was the short word length of the documents in the grooming data. To this end, passing a whole grooming conversation for analysis was tested and although there was variance in the sentiment scores compared to the mean of the total scores for classification of single rows, the results were consistent in a positive sentiment classification.

The use of RDBMS within detecting online grooming discourse proved to have a negative impact on performance when forensically analysing/classifying historic data. Performing the analysis on a database directly proved to be inefficient returning excessive processing times. The research has shown that performing such operations outside of RDBMS speeds up processing exponentially and therefore suggests that any detection system would be better positioned client-side before insertion of data into the intended relational database.

Although the tools and techniques tested can go some way to providing a solution to detecting online predation, there is no “silver bullet” application capable of adapting to the ever-changing lexicon of digital discourse and accurately identifying grooming conversations. Forensic analysis of historic data may prove to be too late in detecting grooming conversation and as this research has shown, the timing of grooming conversations concluding in a meeting vary greatly. Groomers often suggest switching platforms or mediums to maintain the groom outside the initial platform in order to evade detection or to facilitate the transfer of multimedia files to the victim.

Future work would see the combining of existing tools, especially DNN architectures, into one cohesive system for detection. Further development and finer training of the neural network architectures tested during the research would need to take place. Developing a finely tuned pre-processing workflow that maximised the efficiencies and processing of chat corpora would enable more accurate results with the potential for such a workflow to be adapted to other scenarios.

Larger grooming data corpus would enable further analysis of existing grooming trends, evolution in grooming methodology, and changes in the lexicon used by both Predator and Victim. This could lead to identification of triggers in the discourse that could potentially “ring the alarm”.

The on-going development of GPUs and their ever-increasing speed and capability when applied in the field of artificial intelligence, provide possibilities for the creation of real-time solutions required to detect grooming discourse at the earliest opportunity. All the while performing such real-time operations transparently as a background process. The integration of such real-time solutions may also require integration into big data platforms such as Apache SPARK given the huge amount of data generated by chat interactions on a national and global scale. As seen on the UK Channel 4 programme “Undercover Police: Hunting Paedophiles”, groomers often used a scatter gun approach and interacted with a multitude of potential victims concurrently (Channel4, 2021). Therefore, the analysis of a wider field of data than that offered by a client-server application may need to be considered.

As computer scientists and technologists, we have an ethical, legal, and moral responsibility to provide safe platforms for children to exist safely in a digital world that we create. There must be a proactive approach to solving the plight of children online whilst preserving children’s rights to privacy.

7 References

Aggarwal, C., 2018. *Neural Networks and Deep.* 1 ed. New York: Springer.

Al Amrani, Y., Lazaar, M. & El Kadiri, K., 2018. Random Forest and Support Vector Machine based Hybrid Approach to Senitment Analysis. *Procedia Computer Science,* Volume 127, pp. 511-550.

Anderson, J. & Rosendfield, E., 1988. *Neurocomputing: Foundations of Research..* Cambridge: MIT Press.

asciitable.com, 2021. *ASCII Table and Description.* [Online]
Available at: http://www.asciitable.com
[Accessed 28 03 2021].

Baccarini, D. & Mellville, T., 2011. *Risk Management of Research Projects in a University Context - An Exploratory Study.* Australia, Unknown: Bond University ePublications.

Baker, P., 2013. *Visiting With The Brown Family.* [Online]
Available at: http://cass.lancs.ac.uk/visiting-with-the-brown-family/
[Accessed 15 November 2020].

Banks, M. & Card, O. S., 2008. *On the Way to the Web: The Secret History of the Internet and Its Founders..* London: Apress.

Bark.us, 2021. *How.* [Online]
Available at: https://www.bark.us/#how
[Accessed 17 January 2021].

BBC, 2016. *uk-england-leicestershire-36606210.* [Online]
Available at: https://www.bbc.co.uk/news/uk-england-leicestershire-36606210
[Accessed 23 October 2020].

BBC, 2019. *uk-wales-48880998.* [Online]
Available at: https://www.bbc.co.uk/news/uk-wales-48880998
[Accessed 22 October 2020].

Bentley, H. et al., 2020. *How safe are our children?: an overview of data on adolescent abuse,* London: NSPCC.

Beysolow II, T., 2018. *Applied Natural Language Processing with Python Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing.* San Francisco: Apress.

Bilgehan, M., 2011. Comparison of ANFIS and NN models—With a study in critical bucklingload estimation. *Applied Soft Computing,* Volume 11, pp. 3779-3791.

Black, J., Wollis, M., Woodworth, M. & Hancock, J., 2015. A linguistic analysis of grooming strategies of online child sex offenders:. *Child Abuse,* Volume 44, pp. 140-149.

Borj, P. & Bours, P., 2019. *Predatory Conversation Detection.* Doha, s.n., pp. 1-6.

Brezina, V., Well-Tessier, P. & McEnery, A., 2020. *#LancsBox v. 5.x. [software].* [Online]
Available at: http://corpora.lancs.ac.uk/lancsbox/
[Accessed 1 November 2020].

Brittanica, 2020. *who-invented-the-internet.* [Online]
Available at: www.britiannina.com-who-invented-the-internet
[Accessed 20 October 2020].

Cano Basave, A., Fern´andez, M. & Harith, A., 2014. *Detecting child grooming behaviour patterns on social media.* Barcelona, Springer.

Cassandra, 2021. *What is Cassandra?.* [Online]
Available at: https://cassandra.apache.org/
[Accessed 11 3 2021].

Cleanrouter.com, 2021. *Features.* [Online]
Available at: https://cleanrouter.com/features/
[Accessed 16 Janurary 2021].

Codd, E., 1970. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM,* 13(6), pp. 377-387.

Craven, S., S., B. & Gilchrist, E., 2006. Sexual grooming of children : Review of literature and theoretical considerations.. *Journal of Sexual Agression,* Volume 12, pp. 287-299.

Creswell, J. & Plano Clark, V., 2011. *Best Practices for Mixed Methods research in Health Sciences,* Bethesda: Office of Behavioral and Social Sciences Research.

Culpeper, J., 2011. *Impoliteness: Using Language to Cause Offence.* Cambridge: Cambridge University Press.

Damgeti, P., 2017. *Statistics for Machine Learning: Techniques for exploring supervised, unsupervised, and reinforcement learning models with Python and R.* 1 ed. Birmingham: Packt Publishing.

Dowden, O. & Patel, R., 2020. *Online Harms White Paper - Initial consultation response.* [Online]
Available at: https://www.gov.uk/government/consultations/online-harms-white-paper/outcome/online-harms-white-paper-full-government-response
[Accessed 6 February 2021].

Ek, A., Bernardy, J. & Chatzikyriakidis, S., 2020. *How does Punctuation Affect Neural Models in Natural Language.* Gothenburg, Association for Computational Linguistics.

Ertel, W., 2017. *Introduction to Artifical Intelligence.* 2nd ed. Cham: Springer International Publishing.

Frank, E., Hall, M. & Witten, H., 2016. *Data Mining: Practical Machine Learning Tools and Techniques, Fourth Edition.* 4 ed. London: Morgan Kaufmann.

Gov.uk, 2003. *ukpga/2003/42/section/17.* [Online]
Available at: https://www.legislation.gov.uk/ukpga/2003/42/section/17
[Accessed 10 October 2020].

Gov.uk, 2017. *ukpga/2015/9/section/67.* [Online]
Available at: https://www.legislation.gov.uk/ukpga/2015/9/section/67
[Accessed 12 October 2020].

Gupta, A., Sureka, A. & Kumaraguru, P., 2012. Characterizing pedophile conversations on the internet using online grooming. *CoRR,* Volume abs/1208.4324, pp. 1208-4324.

Hadoop, 2021. *Apache Hadoop.* [Online]
Available at: https://hadoop.apache.org/
[Accessed 11 March 2021].

Hafner, K. & Lyon, M., 1996. *Where Wizards Stay Up Late (The Origins Of The Internet).* 1 ed. New York: Simon & Schuster.

Han, J., Kamber, M. & Pei, J., 2012. Getting to Know Your Data. In: M. K. J. P. Jiawei Han, ed. *Data Mining (Third Edition).* London: Morgan Kaufmann, pp. 39-82.

Herring, S., 1997. Computer-mediated discourse analysis: introduction. *Electronic Journal of Communication,* Volume 6, pp. 1-3.

Herring, S., 2010. Computer-Mediated Conversation: Introduction and Overview. *Language@Internet,* Volume 7.

Hochreiter, S. & Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation,* Volume 9, pp. 1735-1780.

Ingram, A., Hathorn, G. & Evans, A., 2000. Beyond chat on the internet. *Computers & Education ,* Volume 35, pp. 21-35.

IWF, 2020. *'definite-jump'-as-hotline-sees-50-increase-public-reports-of-online-child-sexual-abuse-during.* [Online]
Available at: https://www.iwf.org.uk/news/%E2%80%98definite-jump%E2%80%99-as-hotline-sees-50-increase-public-reports-of-online-child-sexual-abuse-during
[Accessed 23 October 2020].

IWF, 2020. *millions-of-attempts-to-access-child-sexual-abuse-online-during-lockdown.* [Online]
Available at: https://www.iwf.org.uk/news/millions-of-attempts-to-access-child-sexual-abuse-online-during-lockdown
[Accessed 23 October 2020].

Johansen, R., Vallee, J. & Spangler, K., 1979. *Electronic Meetings: Technical Alternatives and Social Choices..* Reading, Massachusetts: Addison-Wesley.

Johnson, B., Onwuegbuzie, A. & Turner, L., 2007. Toward a definition of mixed methods research. *Journal of Mixed Methods Research,* Volume 1, pp. 112-133.

Kapersky.co.uk, 2021. *Kapersky Safe Kids.* [Online]
Available at: https://www.kaspersky.co.uk/safe-kids
[Accessed 15 January 2021].

Keerthi Kumar, K., Harish, B. & Darshan, H., 2018. Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method. *International Journal of Interactive Multimedia and Artificial Intelligence,* Volume 5, pp. 109-114.

Keras, 2021. *About.* [Online]
Available at: https://keras.io/about
[Accessed 02 02 2021].

Kholkovskaia, O., 2017. *Role of the Brown Corpus in the History of Corpus.* Praha: Czech Technical University.

Kou, G. et al., 2020. Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. *Applied Soft Computing Journal,* Volume 86.

Ladani, D. & Deasi, N., 2020. *Stopword Identification and Removal Techniques on TC and IR applications: A Survey.* Kondampatti, IEEE.

Lample, G. et al., 2016. *Neural Architectures for Named Entity Recognition.* San Diego, Association for Computational Linguistics.

Lison, P. & Tiedmann, J., 2016. *Extracting Large Corpora from Moivie and TV Subtitles.* Portorož, European Language Resources Association (ELRA).

Lorenzo-Dus, N. & Izura, C., 2017. 'cause ur special'': Understanding trust and complimenting behaviour in online grooming discourse. *Journal of Pragmatics,* Volume 112, pp. 68-82.

Luhn, H., 1958. A Business Intelligence System. *IBM JOURNAL,* pp. 314-319.

Lui, G. & Guo, J., 2019. Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing,* 337(1), pp. 325-338.

Macfarlane, K., 2016. *An Intelligent Multi-Agent System Approach to Automating Safety Features for On-Line Real Time Communications: Agent Mediated Information Exchange,* Huddersfield: University of Huddersfield.

May, C., Cotterall, R. & Van Durme, B., 2019. *An Analysis of Lemmatization on Topic An Analysis of Lemmatization on Topic,* Baltimore: Johns Hopkins University.

McColloch, W. & Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics,* 5(4), pp. 115-133.

Mikolov, T., K, C., Corrado, G. & Dean, J., 2013. *Efficient Estimation of Word Representations in Vector Space.* [Online]
Available at: https://arxiv.org/pdf/1301.3781v3.pdf
[Accessed 14 February 2021].

MongoDB, 2021. *why use mongodb.* [Online]
Available at: https://www.mongodb.com/why-use-mongodb
[Accessed 11 March 2021].

Mustaro, P. & Rossi, R., 2013. *Risk management in scientific research: a proposal guided in Project Management Book of Knowledge and Failure Mode and Effects Analysis.* Oklahoma, IEEE.

Netnanny.com, 2021. *Parental Control.* [Online]
Available at: https://www.netnanny.com/features/parental-controls/
[Accessed 10 January 2021].

NLTK, 2021. *NLTK 3.5 documentation.* [Online]
Available at: https://www.nltk.org/
[Accessed 30 03 2021].

NSPCC, 2019. *over-5000-grooming-offences-recorded-18-months/.* [Online]
Available at: https://www.nspcc.org.uk/about-us/news-opinion/2019/over-5000-grooming-

offences-recorded-18-months/
[Accessed 25 October 2020].

NSPCC, 2020. *wild-west-web.* [Online]
Available at: https://www.nspcc.org.uk/support-us/campaigns/wild-west-web/
[Accessed 10 October 2020].

O'Connell, R., 2003. *A typology of cyber sexploitation and online grooming practices. Preston, England,* Preston: University of Central Lancashire.

Ofcom, 2020. *Children and parents: Media Use and Attitudes Report 2019,* London: OFCOM.

Ofcom, 2020. *Internet users' experience of potential online harms: summary of survey research,* London: Ofcom.

Ólafsson, K. L. S. &. H. L., 2014. *Children's Use of Online Technologies in Europe. A review of the European evidence base,* London: LSE.

Olson, L., Daggs, J., Ellevold, B. & Rogers, T., 2007. Entrapping the Innocent: Toward a Theory of Child Sexual Predators' Luring Communication. *Communication Theory,* Volume 17, pp. 231-251.

ONS, 2020. *Exploring the UK's digital divide.* [Online]
Available at:
https://www.ons.gov.uk/peoplepopulationandcommunity/householdcharacteristics/homeinternetandsocialmediausage/articles/exploringtheuksdigitaldivide/2019-03-04#:~:text=It%20estimates%20that%20the%20number,the%20five%20basic%20digital%20skills).
[Accessed 06 February 2021].

Padurariu, C. & Breaban, M., 2019. Dealing with Data Imbalance in Text Classification. *Procedia Computer Science,* Volume 159, pp. 736-745.

Parapar, J., Losada, D. & A, B., 2012. A learning-based approach for the identification of sexual predators in chat logs. In: P. F. H. P. S. Tiziana Catarci, ed. *Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics.* Rome: Springer.

Pendar, N., 2007. *Toward Spotting the Pedophile Telling victim from predator in text chats.* Santa Monica, IEEE.

Pennington, J., Socher, R. & Manning, C. D., 2014. *GloVe: Global Vectors for Word Representation.* Doha, Association for Computational Linguistics, pp. 313-324.

Qustodio.com, 2020. *qustodio.com.* [Online]
Available at: https://www.qustodio.com
[Accessed 26 October 2020].

Raftery, J., 1996. *Risk Analaysis in Project Management.* 2 ed. London: E & FN SPON.

Reber, P., 2010. *What Is the Memory Capacity of the Human Brain?.* [Online]
Available at: https://www.scientificamerican.com/article/what-is-the-memory-capacity/
[Accessed 25 February 2021].

Reber, P., 2010. *What Is the Memory Capacity of the Human Brain?.* [Online]
Available at: https://www.scientificamerican.com/article/what-is-the-memory-capacity/
[Accessed 22 February 2021].

Sang-Bum, K., Han, K., RIM, H. & Myaeng, S., 2006. Some Effective Techniques for Naive Bayes Some Effective Techniques for Naive Bayes. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,* 18(11), pp. 1457-1466.

Schofield, A. & Minmi, D., 2016. Comparing apples to apple: The effects of stemmers on topic models. *Transactions of the Association for Computational Linguistics,* Volume 4, pp. 287-300.

Seedall, M., Macfarlane, K. & Holmes, V., 2019. *SafeChat System with Natural Language Processing and Deep Neural Networks.* Huddersfield, EMiT/University of Huddersfield/High End Compute Ltd/University of Manchester , pp. 28-31.

Shaoul, C. & Westbury, C., 2019. *A reduced redundancy USENET corpus (2005-2011).* [Online]
Available at: https://www.psych.ualberta.ca/~westburylab/downloads/usenetcorpus.download.html
[Accessed 03 03 2019].

Smith, L., 2017. *Cyclical Learning Rates for Training Neural Networks.* Santa Rosa, U.S. Naval Research Laboratory.

social-media.co.uk, 2020. */list-popular-social-networking-websites.* [Online]
Available at: https://social-media.co.uk/list-popular-social-networking-websites
[Accessed 26 October 2020].

Spärck Jones, K., 1972. A statistical interpretation of term specificity. *Journal of Documentation,* 28(1), pp. 11-21.

Spark, 2021. *Apache Spark.* [Online]
Available at: https://spark.apache.org/
[Accessed 11 3 2021].

Star, D., 2020. */news/latest-news/dad-two-tried-meet-girl-22658327.* [Online]
Available at: https://www.dailystar.co.uk/news/latest-news/dad-two-tried-meet-girl-22658327
[Accessed 23 October 2020].

Statista.com, 2020. *children-ownership-of-tablets-smartphones-by-age-uk/.* [Online]
Available at: https://www.statista.com/statistics/805397/children-ownership-of-tablets-smartphones-by-age-uk/
[Accessed 12 October 2020].

Tensorflow, 2020. *Why TensorFlow.* [Online]
Available at: https://www.tensorflow.org/
[Accessed 12 12 2020].

Tensorflow, 2021. *Word2Vec.* [Online]
Available at: https://www.tensorflow.org/tutorials/text/word2vec
[Accessed 20 03 2021].

tfidf.com, 2021. *http://www.tfidf.com/.* [Online]
Available at: http://www.tfidf.com/
[Accessed 15 November 2020].

Thurlow, C. & Mroczek, K., 2011. *Digital Discourse Language in the New Media.* 1 ed. Oxford: Oxford University Press.

Turing, A., 1936. On Computable Numbers, with an Application to the Entscheidungsproblem. *Uber formal unentscheidbare Satze der Principia Mathematica und ver- Uber formal unentscheidbare Satze der Principia Mathematica und ver-Monatsheftc Math. Phys.,* Volume 38, pp. 173-198.

Turing, A., 1950. Computing Machinery and Intelligence. *Mind,* Volume 59, pp. 433-460.

twitter.com, 2021. *twitter-api.* [Online]
Available at: https://developer.twitter.com/en/docs/twitter-api
[Accessed 18 03 2021].

uk.norton.com, 2021. *Norton Family Premier.* [Online]
Available at: https://uk.norton.com/norton-family-premier
[Accessed 15 January 2021].

UKCCIS, 2015. *Child Safety Online: A Practical Guide for Providers of Social Media and Interactive Services,* London: UKCCIS.

UKCCIS, 2017. *Child Safety Online: A Practical Guide for Providers of Social Media and Interactive Services,* London: UKCCIS.

UNICEF, 2009. *Convention on the Rights of the Child,* London: UNICEF.

UNICEF, 2018. *Children's Online Privacy and freedom of Expression.* London: UNICEF.

Vijayarani, S. & Janani, R., 2016. Text Mining: Open Source Tokenisation Tools - An Analysis. *Advanced Computational Intelligence: An International Journal (ACII),* 3(1), pp. 37-47.

Wang, H. & Castanon, J., 2015. *Sentiment expression via emoticons on social media..* Santa Clara, IEEE.

Webster, J. & Kit, C., 1992. Tokenization as the initial phase in NLP. *Proceedings of the 14th conference on Computational linguistics,* Volume 4, pp. 1106-1110.

Whittle, H. C., Hamilton-Giachritsis, C., Beech, A. & Collings, G., 2013. A Review of Online Grooming: Charateristics and Concerns. *Agression and Violent Behaviour,* Volume 18, pp. 62-70.

Xu, M., David, J. & Kim, S. H., 2018. The Fourth IIndustrial Revolution: Opportunities and Challenges.. *International Journal of Financial Research,* Volume 9, p. 90.

Yin, W., Kann, K., Yu, M. & Schütze, H., 2017. *Comparative Study of CNN and RNN for Natural Language Processing,* Munich: IBM.

zdnet.com, 2012. */article/38-of-kids-on-facebook-are-under-the-minimum-age-of-13/.* [Online]
Available at: https://www.zdnet.com/article/38-of-kids-on-facebook-are-under-the-minimum-age-of-13/
[Accessed 26 October 2020].

Appendices

1: Confirmation request to access Perverted-Justice datacentre and use raw grooming data.

2: List of text emoticon compiled

3: Twitter Data Parser – Java

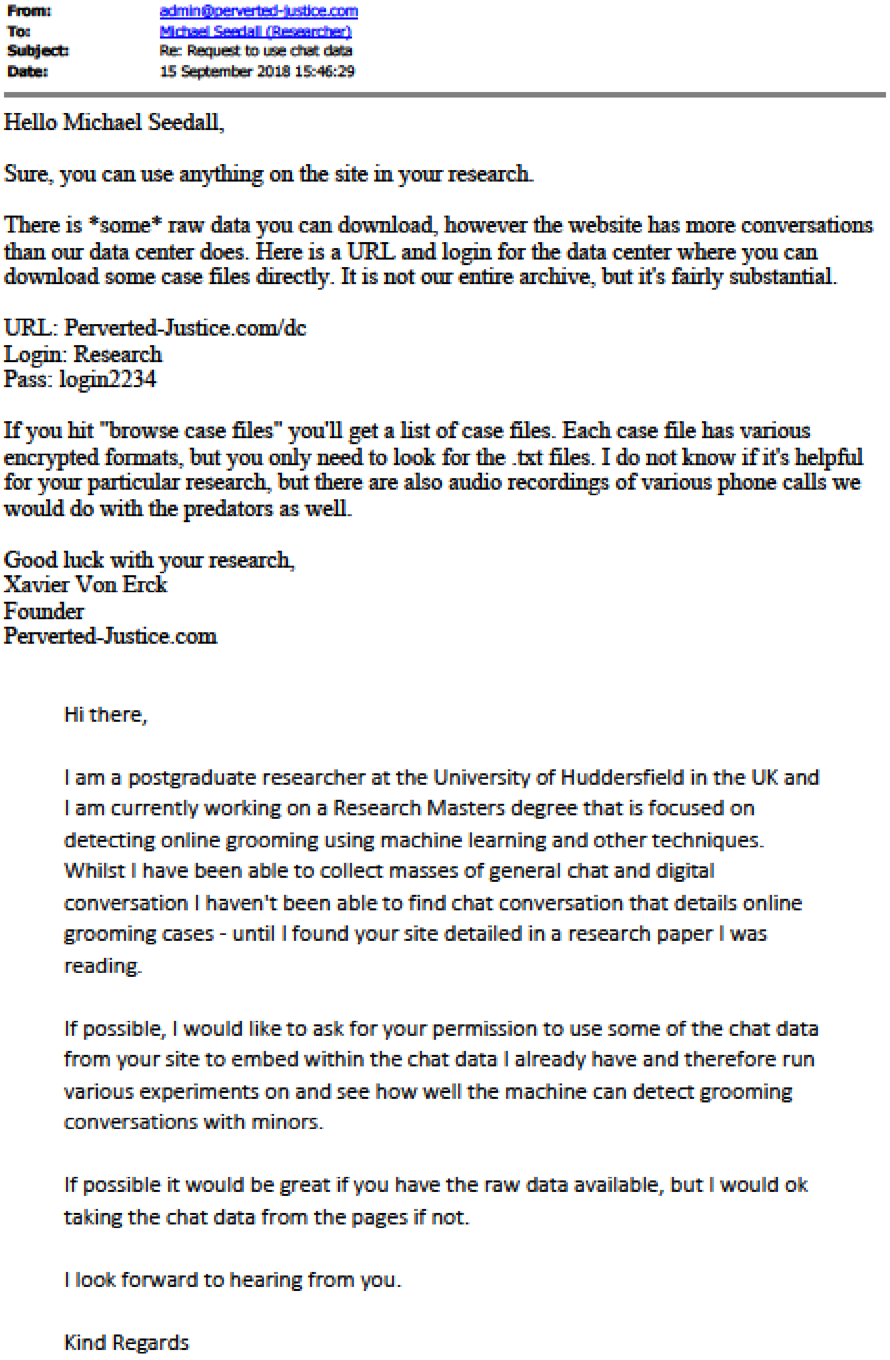
4: Cosine Similarity using TF-IDF

5: Naïve Bayes NLTK Dialogue Classification

6: Feature counts for the search term “babe”

7: Sample analysis report generated by #LancsBox (search terms “mom”, “dad”)

Appendix 1: Confirmation of request for access to Perverted-Justice datacentre and to use raw grooming data for research.



From:	admin@perverted-justice.com
To:	Michael Seedall (Researcher)
Subject:	Re: Request to use chat data
Date:	15 September 2018 15:46:29

Hello Michael Seedall,

Sure, you can use anything on the site in your research.

There is *some* raw data you can download, however the website has more conversations than our data center does. Here is a URL and login for the data center where you can download some case files directly. It is not our entire archive, but it's fairly substantial.

URL: Perverted-Justice.com/dc
Login: Research
Pass: login2234

If you hit "browse case files" you'll get a list of case files. Each case file has various encrypted formats, but you only need to look for the .txt files. I do not know if it's helpful for your particular research, but there are also audio recordings of various phone calls we would do with the predators as well.

Good luck with your research,
Xavier Von Erck
Founder
Perverted-Justice.com

> Hi there,
>
> I am a postgraduate researcher at the University of Huddersfield in the UK and I am currently working on a Research Masters degree that is focused on detecting online grooming using machine learning and other techniques. Whilst I have been able to collect masses of general chat and digital conversation I haven't been able to find chat conversation that details online grooming cases - until I found your site detailed in a research paper I was reading.
>
> If possible, I would like to ask for your permission to use some of the chat data from your site to embed within the chat data I already have and therefore run various experiments on and see how well the machine can detect grooming conversations with minors.
>
> If possible it would be great if you have the raw data available, but I would ok taking the chat data from the pages if not.
>
> I look forward to hearing from you.
>
> Kind Regards

Appendix 2: List of Text Emoticon Compiled

Emoticon	Text Emotion
(.V.)	Alien
O:-)	Angel
X-(	Angry
~:0	Baby
:-D	Big Grin
(*v*)	Bird
:-#	Braces
</3	Broken Heart
=^.^=	Cat
*<:o)	Clown
O.o	Confused
B-)	Cool
:_(	Crying
:'(	Crying
\:D/	Dancing
-	Dazed
:o3	Dog
#-o	Doh!
:*)	Drunk
//_^	Emo
>:)	Evil Grin
<><	Fish
:-(	Frown
:(	Frown
:-(	Frowning
:-P	Frustrated
8-)	Glasses
$_$	Greedy
:->	Grin
=)	Happy
:-)	Happy
:)	Happy
#	Hashtag
<3	Heart
{}	Hug
:-|	Indifferent
X-p	Joking
:-)*	Kiss
:-*	Kiss
:*	Kiss
(-}{-)	Kissing
XD	Laughing

D	Laughing Out Loud
)-:	Left-handed Sad Face
(-:	Left-handed Smiley Face
<3	Love
=/	Mad
:-)(-:	Married
@	Mention
<:3)~	Mouse
~,~	Napping
:-B	Nerd
^_^	Overjoyed
<l:0	Partying
:-/	Perplexed
=8)	Pig
@~)~~~~	Rose
=(	Sad
:-(	Sad
:(	Sad
:S	Sarcastic
:-@	Screaming
:-o	Shocked
:-)	Smile
:)	Smile
:-Q	Smoking
:>	Smug
:P	Sticking Tongue Out
:o	Surprised
:-J	Tongue in Cheek
:-&	Tongue Tied
:-\	Undecided
:-E	Vampire
;-)	Winking
;)	Winking
|-O	Yawn
8-#	Zombie

Appendix 3: Twitter Data Parser

```
import java.io.*;
class TwitterData
{
 public static void main(String args[])
    {
  try{
        // Open the file that is the first
        // command line parameter
        File tDirectory = new File("\\\\UC224-16\\Tutor Share\\Twitter\\TwitterData2\\src\\3");
        File [] tFilesInDir = tDirectory.listFiles();
        for(File tFile: tFilesInDir)
        {
            if(!tFile.isDirectory())
            {
                BufferedReader br = new BufferedReader(new FileReader(tFile));
                String strLine = null;
                PrintWriter out = new PrintWriter(new FileWriter("\\\\UC224-16\\Tutor
Share\\Twitter\\ParsedTwitterOutput.txt",true));
                //Read File Line By Line
                while ((strLine = br.readLine()) != null)
                {
                    //int tStartIndex = strLine.indexOf("\\\"");

                    strLine = strLine.substring(strLine.indexOf("text\":\"") + 7);
                    strLine = strLine.substring(0, strLine.indexOf("\""));
                    //System.out.println(strLine);

                    if(strLine.contains("@")|| strLine.contains("http"))
                    {
                        strLine = strLine.replaceAll("@\\p{L}+", "");
                        strLine = strLine.replaceAll("http\\p{L}+", "");
                    }
                    else
                    {
                        if(strLine.startsWith("RT"))
                        {
                            strLine = strLine.substring(strLine.indexOf(':') 1);
                            int tIndex = strLine.indexOf("\\u");
                            if(tIndex != -1)
                            {
                                strLine = strLine.substring(0,tIndex);
                                System.out.println(strLine);
                                out.println(strLine);
                            }
                        }
                        else if(Character.isUpperCase(strLine.charAt(0))||
strLine.startsWith("@"))
                        {
                            int tIndex = strLine.indexOf("\\u");
                            if(tIndex != -1)
                            {
```

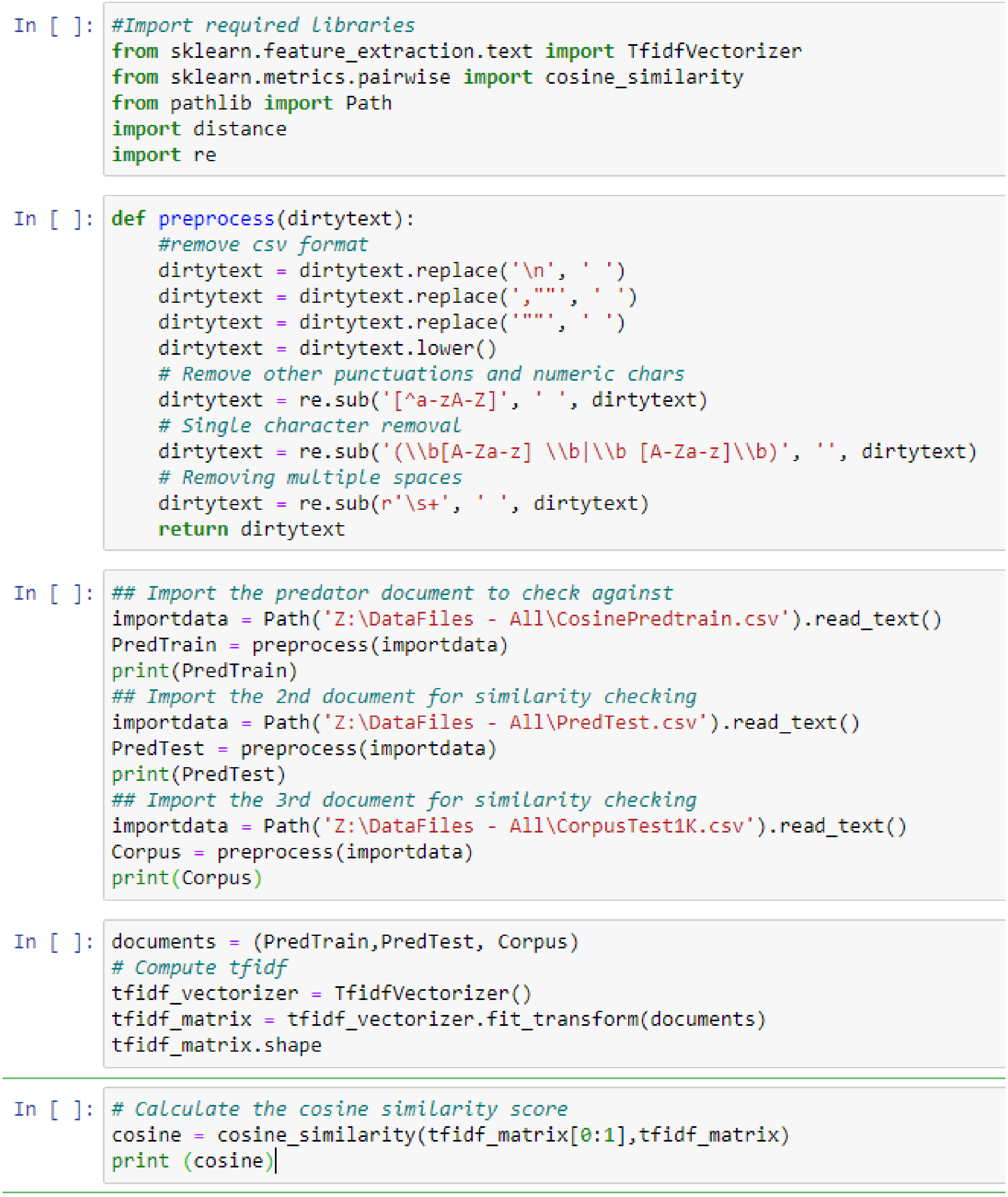
```
                                                            strLine = strLine.substring(0,tIndex);
                                                            System.out.println(strLine);
                                                            out.println(strLine);
                                                    }
                                                    else
                                                    {
                                                            System.out.println(strLine);
                                                            out.println(strLine);
                                                    }
                                            }
                                            else if(strLine.startsWith("\\u"))
                                            {
                                                    continue;
                                            }
                                    }

                }
                br.close();
                out.close();
                        }

                }
                //Close the input stream

                }catch (IOException e){//Catch exception if any
                        System.err.println("Error: " + e.getMessage());
                }
        }
```

Appendix 4 : Cosine Similarity Using TF-IDF



```
In [ ]: #Import required libraries
        from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.metrics.pairwise import cosine_similarity
        from pathlib import Path
        import distance
        import re
```

```
In [ ]: def preprocess(dirtytext):
            #remove csv format
            dirtytext = dirtytext.replace('\n', ' ')
            dirtytext = dirtytext.replace(',""', ' ')
            dirtytext = dirtytext.replace('""', ' ')
            dirtytext = dirtytext.lower()
            # Remove other punctuations and numeric chars
            dirtytext = re.sub('[^a-zA-Z]', ' ', dirtytext)
            # Single character removal
            dirtytext = re.sub('(\\b[A-Za-z] \\b|\\b [A-Za-z]\\b)', '', dirtytext)
            # Removing multiple spaces
            dirtytext = re.sub(r'\s+', ' ', dirtytext)
            return dirtytext
```

```
In [ ]: ## Import the predator document to check against
        importdata = Path('Z:\DataFiles - All\CosinePredtrain.csv').read_text()
        PredTrain = preprocess(importdata)
        print(PredTrain)
        ## Import the 2nd document for similarity checking
        importdata = Path('Z:\DataFiles - All\PredTest.csv').read_text()
        PredTest = preprocess(importdata)
        print(PredTest)
        ## Import the 3rd document for similarity checking
        importdata = Path('Z:\DataFiles - All\CorpusTest1K.csv').read_text()
        Corpus = preprocess(importdata)
        print(Corpus)
```

```
In [ ]: documents = (PredTrain,PredTest, Corpus)
        # Compute tfidf
        tfidf_vectorizer = TfidfVectorizer()
        tfidf_matrix = tfidf_vectorizer.fit_transform(documents)
        tfidf_matrix.shape
```

```
In [ ]: # Calculate the cosine similarity score
        cosine = cosine_similarity(tfidf_matrix[0:1],tfidf_matrix)
        print (cosine)
```

Appendix 5 : Naïve Bayes NLTK Dialogue Classification

```
import nltk
import mysql.connector
def dialogue_act_features(post):
  features = {}
  for word in nltk.word_tokenize(post):
    features['contains({})'.format(word.lower())] = True
  return features
posts = nltk.corpus.nps_chat.xml_posts()[:10000000]
featuresets = [(dialogue_act_features(post.text), post.get('class')) for post in posts]
size = int(len(featuresets) * 0.2)
train_set, test_set = featuresets[size:], featuresets[:size]
classifier = nltk.NaiveBayesClassifier.train(train_set)
print(nltk.classify.accuracy(classifier, test_set))
mydb = mysql.connector.connect(host='ucbccluster.blackburn.ac.uk',
                               database='chat_research',
                               user='+++++++++++',
                               password='+++++++++',use_pure=True)
cur = mydb.cursor()
statement = "SELECT ID, ChatTextWithNoEmoji FROM tblPredParsed ORDER BY ID"
cur.execute(statement)
for row in cur.fetchall():
   strtext = str(row[1]).lower()
   classy = classifier.classify(dialogue_act_features(strtext))
   cur = mydb.cursor()
   updater = "UPDATE tblPredParsed SET Interrogative = %s WHERE ID = %s"
   input = (classy, row[0])
   cur.execute(updater, input)
   mydb.commit()
```

Appendix 6 : Feature Counts for the word babe.

qryFeatureCountAnalysis			
Token	**PredVictim**	**Reddit**	**Westbury Corpus**
babeactually	1	0	0
babebabewhatchy	1	0	0
babebabydshes	1	0	0
babebackstupid	1	0	0
babebad	1	0	0
babebiggxyou	1	0	0
babebout	1	0	0
babebye	1	0	0
babecan	1	0	0
babecant	2	0	0
babedapplying	1	0	0
babedidgot	1	0	0
babedont	2	0	0
babedu	1	0	0
babedush	1	0	0
babeeee	1	0	0
babeeeee	2	0	0
babeeeeeee	1	0	0
babeeeeeeu	1	0	0
babeget	1	0	0
babegetting	3	0	0
babegoodhihistupid	2	0	0
babegot	2	0	0
babeguna	2	0	0
babeguy	1	0	0
babehaha	2	0	0
babehahahaha	1	0	0
babeheresilly	1	0	0
babehey	3	0	0
babeheya	1	0	0
babeheynothing	1	0	0
babeheyuwatching	1	0	0
babehishis	1	0	0
babehisim	1	0	0
babehisnothing	1	0	0
babehiswhatchy	1	0	0
babehiwatgot	1	0	0
babeholy	1	0	0
babehope	3	0	0

qryFeatureCountAnalysis

Token	PredVictim	Reddit	Westbury Corpus
babehow	1	0	0
babehowwho	1	0	0
babehuhwww	1	0	0
babehushtylove	1	0	0
babeim	4	0	0
babeink	2	0	0
babejus	2	0	0
babek	1	0	0
babekklove	1	0	0
babekkyep	1	0	0
babeklove	1	0	0
babeknow	1	0	0
babekya	1	0	0
babekyepmummdwho	1	0	0
babelike	1	0	0
babelong	1	0	0
babelove	8	0	0
babelow	1	0	0
babemay	2	0	0
babemiss	1	0	0
babemissed	1	0	0
babemoneymen	1	0	0
babemumm	1	0	0
babenever	1	0	0
babenite	1	0	0
babeno	1	0	0
babeoh	3	0	0
babeohwho	1	0	0
babeok	1	0	0
babeorbbackthought	1	0	0
babeorbk	1	0	0
babeorg	1	0	0
babeplain	1	0	0
babeply	1	0	0
baber	1	0	0
babereasonwho	1	0	0
babeseen	1	0	0
babeshocku	1	0	0
babeshortsplain	1	0	0
babesorry	1	0	0

qryFeatureCountAnalysis

Token	PredVictim	Reddit	Westbury Corpus
babesup	2	0	0
babesweet	2	0	0
babetaste	1	0	0
babethe	1	0	0
babeu	19	0	0
babeur	1	0	0
babeus	1	0	0
babeutv	1	0	0
babeutvim	1	0	0
babeuya	1	0	0
babewant	2	0	0
babewaynoolove	1	0	0
babewehee	2	0	0
babewell	1	0	0
babewho	4	0	0
babewht	2	0	0
babewhy	1	0	0
babewish	1	0	0
babewishu	1	0	0
babewondering	1	0	0
babewont	1	0	0
babewuna	1	0	0
babewunna	1	0	0
babeya	7	0	0
babeyep	2	0	0
babeyesabsolutely	2	0	0
bebabema	1	0	0
byeebabe	2	0	0
carefulbabe	2	0	0
churchbabe	1	0	0
coolbabe	1	0	0
coolmummyahuhbabego	1	0	0
crybabe	1	0	0
dildobabe	1	0	0
fridaybabe	1	0	0
gnitehiiiiiiiihellllloooooooooooobabeeeeeer	1	0	0
gohuhohbabe	1	0	0
longbabe	1	0	0
lovebabeim	1	0	0
lowbabe	1	0	0

qryFeatureCountAnalysis

Token	PredVictim	Reddit	Westbury Corpus
mustardbabe	1	0	0
nitebabe	1	0	0
pbabe	1	0	0
plybabe	1	0	0
publicbabe	2	0	0
rbabe	1	0	0
remorebabe	1	0	0
sisokbabe	1	0	0
sisokbaber	1	0	0
somethingbabe	1	0	0
stillbabeu	1	0	0
thatweheebabe	1	0	0
thebabe	1	0	0
thebabebabe	1	0	0
thembabe	1	0	0
tiredbabe	1	0	0
todaybabe	1	0	0
todaybabeaw	1	0	0
toobabe	1	0	0
ubabe	3	0	0
ubabebabehiwent	1	0	0
ubabehiim	1	0	0
ubabehiseriouslyu	1	0	0
ubabewatching	1	0	0
unitehishulloobabetime	1	0	0
uxbabe	1	0	0
waitbabe	1	0	0
weekendgoodyababe	1	0	0

Appendix 7: Sample analysis report generated by #LancsBox (search terms "mom", "dad")

Introduction

This research report was automatically produced by #LancsBox (Brezina et al., 2015, 2018, 2020), a corpus analysis tool developed at Lancaster University. It uses cutting-edge technology and statistical sophistication (Brezina 2018) to analyze and visualize corpus data. For more information and tips on research report writing see the Research Report Guide.

Method

Data

The study analyzed the following corpus:

Table 1. Corpus used

Name	Language	Texts	Tokens	Additional information
Corpus 1	English	1	213,501	Types: 11,538 Lemmas: 11,740

In the study, 1 corpus was used of the total size of 213,501 running words (tokens) in 1 texts. A full description of the corpora is available in data\tsv\corpora.

Procedure

#LancsBox (Brezina et al., 2015, 2018, 2020) software package was employed to analyse the data. The following tools from the package were used: KWIC, GraphColl, Whelk, Words, Ngrams and Text. The KWIC tool generates a list of all instances of a search term in a corpus in the form of a concordance. The GraphColl tool identifies collocations and displays them in a table and as a collocation graph or network. The Whelk tool provides information about how the search term is distributed across corpus files. The Words tool allows in-depth analysis of frequencies of types, lemmas and POS categories as well as comparison of corpora using the keywords technique. The Ngrams tool allows in-depth analysis of frequencies of ngram types, lemmas and POS categories as well as comparison of corpora using the key ngram technique. The Text tool enables an in-depth insight into the context in which a word or phrase is used. The following search terms were used: "mom" and "dad".

Results

General overview: Frequency lists

Table 2 shows the frequencies (both absolute and relative) and dispersions (CV) of the top ten types in the selected corpora. Longer frequency lists are available in the Appendix (Table 12). The full data is available in data\tsv\words.

Table 2. Top ten types in Corpus 1

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
1	u	15302 (716.718)	0
2	ok	3802 (178.079)	0

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
3	like	3447 (161.451)	0
4	want	2937 (137.564)	0
5	low	2861 (134.004)	0
6	would	2391 (111.990)	0
7	im	2382 (111.569)	0
8	know	2308 (108.103)	0
9	get	2166 (101.452)	0
10	ur	2039 (95.503)	0

Table 3 shows the frequencies (both absolute and relative) and dispersions (CV) of the top ten ngram types in the selected corpora. Longer frequency lists are available in the Appendix (Table 13). The full data is available in data\tsv\ngrams.

Table 3. Top ten ngram types in Corpus 1

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
1	u want	930 (43.560)	0
2	u like	717 (33.583)	0
3	u u	540 (25.293)	0
4	would u	455 (21.311)	0
5	r u	419 (19.625)	0
6	u know	386 (18.080)	0
7	ok u	336 (15.738)	0
8	u think	323 (15.129)	0
9	see u	311 (14.567)	0
10	u wanna	302 (14.145)	0

Specific searches: Concordances and contexts

Search term "mom" in Corpus 1

The search term *mom* occurs 441 times (20.656 per 10k) in Corpus 1 in 1 out of 1 texts. The distribution of this search term in the individual texts can be seen in Table 4. Table 5 displays a random sample of 10 concordance lines, showing the most immediate contexts in which the search term is used. Table 6 shows the use of the search term in a broader context.

Table 4. Distribution of the search term *mom* in Corpus 1

File	Tokens	Frequency	Relative frequency per 10k
PredatorOnly.csv	213501	441	20.656

Table 5. A random set of concordance lines for *mom* in Corpus 1

Filename	Left	Node	Right
PredatorOnly.csv	want see one cool think	mom	would let come over yea
PredatorOnly.csv	at could hang house what	mom	ic would u fool around
PredatorOnly.csv	yes indeed u like ur	mom	working days happens nite u
PredatorOnly.csv	[31mn dad [31mok [31mu sleep	mom	wen dad home ur room
PredatorOnly.csv	horny probably maybe k ur	mom	near wut r u wearing
PredatorOnly.csv	tha jus remember the ur	mom	help u u know like
PredatorOnly.csv	mature [31mme* [31mdid u r	mom	dad help u talking teen
PredatorOnly.csv	theyre cute low thats something	mom	would say friend cool old
PredatorOnly.csv	spank u u maybe tell	mom	d trying keep u pure
PredatorOnly.csv	smart familythats good u live	mom	full time u ever there

Table 6. Random example of the use of *mom* in a broader context in Corpus 1

Context
im curious want see one cool think mom would let come over yea ur house

Search term "dad" in Corpus 1

The search term *dad* occurs 247 times (11.569 per 10k) in Corpus 1 in 1 out of 1 texts. The distribution of this search term in the individual texts can be seen in Table 7. Table 8 displays a random sample of 10 concordance lines, showing the most immediate contexts in which the search term is used. Table 9 shows the use of the search term in a broader context.

Table 7. Distribution of the search term *dad* in Corpus 1

File	Tokens	Frequency	Relative frequency per 10k
PredatorOnly.csv	213501	247	11.569

Table 8. A random set of concordance lines for *dad* in Corpus 1

Filename	Left	Node	Right
PredatorOnly.csv	brain map low absolutely sure	dad	wont back ok thats cool
PredatorOnly.csv	hi hi nothing k ur	dad	gone good phone one sec
PredatorOnly.csv	unless think nope yeah would	dad	husband want niece meant uncle
PredatorOnly.csv	sure what ok think im	dad	ok need run minutes here
PredatorOnly.csv	even want sex me.......and concensual.......and	dad	find go jail long time
PredatorOnly.csv	brother stood till saturday helped	dad	fix busted van kept busy
PredatorOnly.csv	talking teen age time [31mmom	dad	[31mthats good [31mdo u like
PredatorOnly.csv	always win ever sit talk	dad	one one around grams like
PredatorOnly.csv	miss u nothing ate food	dad	nothing yea u today thats
PredatorOnly.csv	dad [31mhe must b nice	dad	[31mwhat r u besides chatting?

Table 9. Random example of the use of *dad* in a broader context in Corpus 1

Context
yeah brain map
low
absolutely sure dad wont back
ok thats cool
yup

Word associations: Collocations

Tables 10 - 11 show the top 10 collocates of *dad* and *mom* in Corpus 1 identified using the Collocation frequency (01 - Freq (5.0), L5-R5, C: 5.0-NC: 5.0). Figure 1 displays a collocation network for all search terms in the same graph. More extensive lists of collocates are available in the Appendix (Tables 14 - 15). The full data is available in data\tsv\graphColl.

Table 10. Collocates of the search term *dad* in Corpus 1

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
1	R	u	157	157	15302
2	L	ur	77	77	2039
3	R	ok	58	58	3802
4	L	mom	52	52	441
5	R	home	36	36	488
6	R	get	31	31	2166
7	L	like	29	29	3447
8	L	im	27	27	2382
9	L	want	27	27	2937
10	L	low	25	25	2861

Table 11. Collocates of the search term *mom* in Corpus 1

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
1	R	u	291	291	15302
2	L	ur	184	184	2039
3	R	ok	89	89	3802
4	R	home	68	68	488
5	R	oh	62	62	1293
6	R	cool	53	53	1294

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
7	R	dad	52	52	247
8	R	get	45	45	2166
9	L	low	44	44	2861
10	R	work	41	41	620

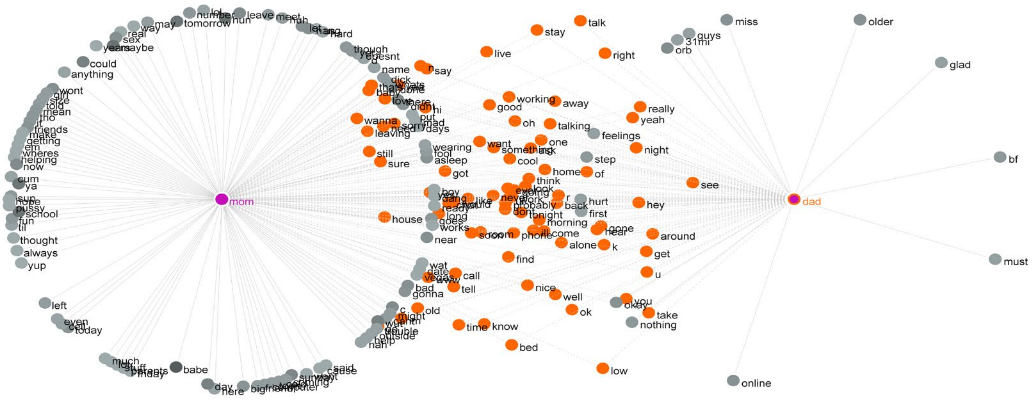


Figure 1. Collocation network: *mom* and *dad* in Corpus 1 (01 - Freq (5.0), L5-R5, C: 5.0-NC: 5.0)

In Figure 1, 185 collocates of *mom* and 99 collocates of *dad* have been displayed. There are 82 shared collocates (*alone*, *around*, *ask*, *away*, *baby*, *back*, *bed*, *call*, *chat*, *come*, *cool*, *dont*, *ever*, *find*, *get*, *go*, *going*, *gone*, *good*, *got*, *hear*, *hey*, *hi*, *home*, *house*, *ill*, *im*, *k*, *know*, *leaving*, *like*, *live*, *long*, *look*, *love*, *low*, *morning*, *n*, *need*, *never*, *nice*, *night*, *of*, *oh*, *ok*, *old*, *one*, *phone*, *probably*, *r*, *really*, *right*, *room*, *say*, *see*, *something*, *soon*, *sorry*, *stay*, *still*, *sure*, *take*, *talk*, *talking*, *tell*, *thats*, *think*, *time*, *tonight*, *u*, *ur*, *wanna*, *want*, *well*, *work*, *working*, *would*, *www*, *yea*, *yeah*, *yes* and *you*). A full table with shared collocates' details is available in the Appendix (Table 16). The full data is available in data\tsv\graphColl\sharedGraphColl_001.tsv.

Statistical analysis

So far, descriptive statistical analysis was reported in the sections above. This includes the analysis of frequency and dispersion (sections 3.1 and 3.2) and collocations (section 3.3). More details about these procedures can be found in Brezina (2018).

No further statistical analysis was carried out because no suitable corpus with enough sampling points (texts) was selected.

References

Brezina, V., Weill-Tessier, P., & McEnery, T. (2020). #LancsBox 5.x [software]. Available at: http://corpora.lancs.ac.uk/lancsbox.

Brezina, V., Timperley, M., & McEnery, T. (2018). #LancsBox 4.x [software]. Available at: http://corpora.lancs.ac.uk/lancsbox.

Brezina, V., McEnery, T., & Wattam, S. (2015). Collocations in context: A new perspective on collocation networks. *International Journal of Corpus Linguistics, 20*(2), 139-173.

Brezina, V. (2018). *Statistics in corpus linguistics: A practical guide*. Cambridge: Cambridge University Press.

Appendix

Table 12. Types in Corpus 1 - (Truncated [100])

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
1	u	15302 (716.718)	0
2	ok	3802 (178.079)	0
3	like	3447 (161.451)	0
4	want	2937 (137.564)	0
5	low	2861 (134.004)	0
6	would	2391 (111.990)	0
7	im	2382 (111.569)	0
8	know	2308 (108.103)	0
9	get	2166 (101.452)	0
10	ur	2039 (95.503)	0
11	yes	1969 (92.224)	0
12	good	1725 (80.796)	0
13	well	1602 (75.035)	0
14	see	1600 (74.941)	0
15	baby	1417 (66.370)	0
16	love	1386 (64.918)	0
17	go	1310 (61.358)	0
18	think	1308 (61.264)	0
19	dont	1298 (60.796)	0
20	cool	1294 (60.609)	0
21	oh	1293 (60.562)	0
22	yeah	1260 (59.016)	0
23	babe	1249 (58.501)	0
24	wanna	1084 (50.773)	0
25	really	1057 (49.508)	0
26	one	1029 (48.196)	0

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
27	time	961 (45.011)	0
28	got	911 (42.670)	0
29	thats	901 (42.201)	0
30	tell	895 (41.920)	0
31	yea	886 (41.499)	0
32	could	861 (40.328)	0
33	going	819 (38.360)	0
34	make	818 (38.314)	0
35	talk	802 (37.564)	0
36	r	791 (37.049)	0
37	right	778 (36.440)	0
38	call	777 (36.393)	0
39	ya	775 (36.300)	0
40	ill	771 (36.112)	0
41	sure	728 (34.098)	0
42	k	718 (33.630)	0
43	sorry	718 (33.630)	0
44	back	682 (31.944)	0
45	gonna	670 (31.382)	0
46	nice	653 (30.585)	0
47	feel	646 (30.257)	0
48	hi	635 (29.742)	0
49	work	620 (29.040)	0
50	much	617 (28.899)	0
51	cant	588 (27.541)	0
52	let	586 (27.447)	0
53	ever	585 (27.400)	0
54	take	584 (27.354)	0

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
55	anything	581 (27.213)	0
56	come	575 (26.932)	0
57	you	553 (25.902)	0
58	hey	550 (25.761)	0
59	day	536 (25.105)	0
60	mm	534 (25.012)	0
61	hun	533 (24.965)	0
62	girl	530 (24.824)	0
63	say	523 (24.496)	0
64	maybe	521 (24.403)	0
65	lol	512 (23.981)	0
66	something	512 (23.981)	0
67	fun	509 (23.841)	0
68	try	506 (23.700)	0
69	still	498 (23.325)	0
70	home	488 (22.857)	0
71	thinking	483 (22.623)	0
72	never	480 (22.482)	0
73	meet	471 (22.061)	0
74	long	467 (21.873)	0
75	sexy	452 (21.171)	0
76	way	450 (21.077)	0
77	night	443 (20.749)	0
78	mom	441 (20.656)	0
79	hope	440 (20.609)	0
80	sex	436 (20.421)	0
81	kiss	435 (20.375)	0
82	wish	435 (20.375)	0

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
83	tomorrow	434 (20.328)	0
84	me	428 (20.047)	0
85	school	425 (19.906)	0
86	need	419 (19.625)	0
87	look	414 (19.391)	0
88	mean	405 (18.969)	0
89	www	403 (18.876)	0
90	cum	401 (18.782)	0
91	sweet	397 (18.595)	0
92	around	388 (18.173)	0
93	pic	386 (18.080)	0
94	ask	382 (17.892)	0
95	wait	378 (17.705)	0
96	bad	372 (17.424)	0
97	n	368 (17.236)	0
98	okay	368 (17.236)	0
99	pics	368 (17.236)	0
100	guy	366 (17.143)	0

Table 13. Ngram types in Corpus 1 - (Truncated [100])

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
1	u want	930 (43.560)	0
2	u like	717 (33.583)	0
3	u u	540 (25.293)	0
4	would u	455 (21.311)	0
5	r u	419 (19.625)	0
6	u know	386 (18.080)	0
7	ok u	336 (15.738)	0
8	u think	323 (15.129)	0

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
9	see u	311 (14.567)	0
10	u wanna	302 (14.145)	0
11	u r	300 (14.052)	0
12	u get	298 (13.958)	0
13	want u	270 (12.646)	0
14	like u	262 (12.272)	0
15	u dont	248 (11.616)	0
16	oh ok	247 (11.569)	0
17	love u	243 (11.382)	0
18	know u	237 (11.101)	0
19	u would	226 (10.585)	0
20	u ever	208 (9.742)	0
21	ok ok	207 (9.696)	0
22	would like	202 (9.461)	0
23	cant wait	196 (9.180)	0
24	low u	193 (9.040)	0
25	yes u	191 (8.946)	0
26	think u	181 (8.478)	0
27	im sorry	179 (8.384)	0
28	u got	177 (8.290)	0
29	u ok	175 (8.197)	0
30	dont know	171 (8.009)	0
31	well u	166 (7.775)	0
32	ur mom	156 (7.307)	0
33	dont want	153 (7.166)	0
34	u tell	153 (7.166)	0
35	tell u	152 (7.119)	0
36	wanna see	149 (6.979)	0

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
37	u call	144 (6.745)	0
38	babe babe	142 (6.651)	0
39	good u	141 (6.604)	0
40	u gonna	139 (6.511)	0
41	u see	139 (6.511)	0
42	cool u	138 (6.464)	0
43	ok baby	136 (6.370)	0
44	make u	132 (6.183)	0
45	u going	131 (6.136)	0
46	im gonna	130 (6.089)	0
47	hi hi	127 (5.948)	0
48	talk u	126 (5.902)	0
49	u go	126 (5.902)	0
50	want see	125 (5.855)	0
51	low low	121 (5.667)	0
52	low ok	121 (5.667)	0
53	thats good	118 (5.527)	0
54	wish could	118 (5.527)	0
55	love babe	115 (5.386)	0
56	let know	113 (5.293)	0
57	im sure	112 (5.246)	0
58	u wearing	112 (5.246)	0
59	yes baby	112 (5.246)	0
60	hope u	111 (5.199)	0
61	u could	111 (5.199)	0
62	u live	110 (5.152)	0
63	wish u	110 (5.152)	0
64	u love	109 (5.105)	0

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
65	kiss u	107 (5.012)	0
66	oh yes	106 (4.965)	0
67	u feel	106 (4.965)	0
68	u really	106 (4.965)	0
69	sure u	105 (4.918)	0
70	baby u	100 (4.684)	0
71	ok hun	99 (4.637)	0
72	call u	98 (4.590)	0
73	thats cool	98 (4.590)	0
74	u good	98 (4.590)	0
75	baby girl	97 (4.543)	0
76	time u	96 (4.496)	0
77	u baby	96 (4.496)	0
78	u ur	93 (4.356)	0
79	babe love	91 (4.262)	0
80	feel like	91 (4.262)	0
81	would love	91 (4.262)	0
82	yes babe	91 (4.262)	0
83	get trouble	90 (4.215)	0
84	yea u	90 (4.215)	0
85	last night	89 (4.169)	0
86	ok well	89 (4.169)	0
87	ok cool	88 (4.122)	0
88	would want	88 (4.122)	0
89	u yes	87 (4.075)	0
90	ok im	86 (4.028)	0
91	make sure	83 (3.888)	0
92	get u	81 (3.794)	0

ID	Type	Absolute frequency (Relative frequency)	Dispersion (CV)
93	im going	81 (3.794)	0
94	go bed	80 (3.747)	0
95	u let	79 (3.700)	0
96	want know	79 (3.700)	0
97	feel good	78 (3.653)	0
98	miss u	78 (3.653)	0
99	low well	77 (3.607)	0
100	u sure	77 (3.607)	0

Table 14. Collocates of the search term *dad* in Corpus 1

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
1	R	u	157	157	15302
2	L	ur	77	77	2039
3	R	ok	58	58	3802
4	L	mom	52	52	441
5	R	home	36	36	488
6	R	get	31	31	2166
7	L	like	29	29	3447
8	L	im	27	27	2382
9	L	want	27	27	2937
10	L	low	25	25	2861
11	R	oh	25	25	1293
12	L	good	23	23	1725
13	L	sorry	20	20	718
14	L	well	20	20	1602
15	L	yeah	19	19	1260
16	R	time	19	19	961
17	L	one	18	18	1029
18	R	know	17	17	2308

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
19	L	go	17	17	1310
20	L	sure	15	15	728
21	L	think	15	15	1308
22	R	would	14	14	2391
23	R	old	14	14	306
24	R	see	14	14	1600
25	R	cool	14	14	1294
26	L	got	14	14	911
27	R	thats	13	13	901
28	R	work	12	12	620
29	R	really	12	12	1057
30	R	come	11	11	575
31	R	dont	11	11	1298
32	R	going	10	10	819
33	R	long	10	10	467
34	M	call	10	10	777
35	L	tell	10	10	895
36	R	of	10	10	95
37	R	room	10	10	153
38	L	ever	9	9	585
39	L	right	9	9	778
40	R	r	9	9	791
41	R	gone	9	9	133
42	L	house	9	9	274
43	R	around	9	9	388
44	R	alone	9	9	258
45	L	live	9	9	292
46	L	yes	9	9	1969
47	R	night	8	8	443

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
48	R	k	8	8	718
49	M	dad	8	8	247
50	R	find	8	8	232
51	L	orb	8	8	126
52	L	baby	8	8	1417
53	L	soon	7	7	269
54	L	okay	7	7	368
55	R	wanna	7	7	1084
56	L	talking	7	7	327
57	R	older	7	7	247
58	R	ask	7	7	382
59	L	tonight	7	7	337
60	L	talk	7	7	802
61	L	look	7	7	414
62	L	back	7	7	682
63	L	miss	7	7	220
64	L	online	7	7	140
65	R	hey	6	6	550
66	M	still	6	6	498
67	L	nothing	6	6	265
68	L	love	6	6	1386
69	R	bf	6	6	183
70	M	probably	6	6	222
71	L	n	6	6	368
72	M	nice	6	6	653
73	M	bed	6	6	352
74	R	never	6	6	480
75	R	take	6	6	584
76	R	leaving	6	6	74

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
77	R	yea	6	6	886
78	M	working	6	6	151
79	L	step	6	6	15
80	L	hi	5	5	635
81	R	you	5	5	553
82	R	something	5	5	512
83	L	morning	5	5	165
84	R	phone	5	5	355
85	R	must	5	5	92
86	L	hurt	5	5	122
87	L	31mi	5	5	106
88	R	stay	5	5	223
89	L	chat	5	5	234
90	L	feelings	5	5	30
91	L	first	5	5	349
92	R	www	5	5	403
93	R	need	5	5	419
94	R	ill	5	5	771
95	L	guys	5	5	266
96	L	away	5	5	221
97	R	hear	5	5	206
98	L	say	5	5	523
99	R	glad	5	5	132

Table 15. Collocates of the search term *mom* in Corpus 1 - (Truncated [100])

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
1	R	u	291	291	15302
2	L	ur	184	184	2039
3	R	ok	89	89	3802
4	R	home	68	68	488

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
5	R	oh	62	62	1293
6	R	cool	53	53	1294
7	R	dad	52	52	247
8	R	get	45	45	2166
9	L	low	44	44	2861
10	R	work	41	41	620
11	L	like	41	41	3447
12	M	im	40	40	2382
13	R	know	39	39	2308
14	M	would	38	38	2391
15	L	want	38	38	2937
16	R	go	37	37	1310
17	L	babe	36	36	1249
18	L	see	35	35	1600
19	L	good	34	34	1725
20	L	time	33	33	961
21	L	one	31	31	1029
22	L	yea	31	31	886
23	L	k	28	28	718
24	M	well	28	28	1602
25	L	dont	26	26	1298
26	R	call	25	25	777
27	L	sorry	25	25	718
28	L	hi	22	22	635
29	L	yes	22	22	1969
30	R	hun	21	21	533
31	R	still	21	21	498
32	R	going	21	21	819
33	R	come	21	21	575

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
34	R	got	21	21	911
35	R	talk	21	21	802
36	M	yeah	20	20	1260
37	L	maybe	20	20	521
38	L	think	20	20	1308
39	L	school	20	20	425
40	R	gone	20	20	133
41	L	ya	19	19	775
42	L	tomorrow	19	19	434
43	L	tell	19	19	895
44	M	mom	18	18	441
45	R	back	18	18	682
46	R	ever	17	17	585
47	L	long	17	17	467
48	L	right	17	17	778
49	R	bed	17	17	352
50	R	tonight	17	17	337
51	L	baby	17	17	1417
52	R	wanna	15	15	1084
53	L	something	15	15	512
54	R	alone	15	15	258
55	L	thats	14	14	901
56	R	wut	14	14	208
57	L	could	14	14	861
58	M	day	14	14	536
59	R	working	14	14	151
60	L	r	13	13	791
61	R	ask	13	13	382
62	L	around	13	13	388

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
63	R	sunday	12	12	73
64	L	live	12	12	292
65	R	there	12	12	177
66	L	sure	11	11	728
67	L	love	11	11	1386
68	R	c	11	11	145
69	L	nice	11	11	653
70	R	leave	11	11	230
71	R	find	11	11	232
72	L	now	11	11	180
73	R	bad	10	10	372
74	R	let	10	10	586
75	R	near	10	10	81
76	L	old	10	10	306
77	R	night	10	10	443
78	R	hang	10	10	204
79	L	sex	10	10	436
80	L	wont	10	10	283
81	L	look	10	10	414
82	M	house	10	10	274
83	R	say	10	10	523
84	R	room	10	10	153
85	R	really	10	10	1057
86	R	big	9	9	354
87	R	asleep	9	9	61
88	L	phone	9	9	355
89	R	didnt	9	9	190
90	R	gonna	9	9	670
91	R	n	9	9	368

ID	Position	Collocate	Stat (Freq)	Freq coll	Freq corpus
92	L	friends	9	9	263
93	R	never	9	9	480
94	L	today	9	9	329
95	M	number	8	8	152
96	L	it	8	8	338
97	L	sup	8	8	48
98	R	though	8	8	302
99	L	chat	8	8	234
100	L	tho	8	8	126

Table 16. Shared collocates for *mom* and *dad* in Corpus 1

ID	Term	FREQ (CORPUS)	NO OF NODES	NODES
1	alone	258	2	dad, mom
2	around	388	2	dad, mom
3	ask	382	2	dad, mom
4	away	221	2	dad, mom
5	baby	1417	2	dad, mom
6	back	682	2	dad, mom
7	bed	352	2	dad, mom
8	call	777	2	dad, mom
9	chat	234	2	dad, mom
10	come	575	2	dad, mom
11	cool	1294	2	dad, mom
12	dont	1298	2	dad, mom
13	ever	585	2	dad, mom
14	find	232	2	dad, mom
15	get	2166	2	dad, mom
16	go	1310	2	dad, mom
17	going	819	2	dad, mom
18	gone	133	2	dad, mom

ID	Term	FREQ (CORPUS)	NO OF NODES	NODES
19	good	1725	2	dad, mom
20	got	911	2	dad, mom
21	hear	206	2	dad, mom
22	hey	550	2	dad, mom
23	hi	635	2	dad, mom
24	home	488	2	dad, mom
25	house	274	2	dad, mom
26	ill	771	2	dad, mom
27	im	2382	2	dad, mom
28	k	718	2	dad, mom
29	know	2308	2	dad, mom
30	leaving	74	2	dad, mom
31	like	3447	2	dad, mom
32	live	292	2	dad, mom
33	long	467	2	dad, mom
34	look	414	2	dad, mom
35	love	1386	2	dad, mom
36	low	2861	2	dad, mom
37	morning	165	2	dad, mom
38	n	368	2	dad, mom
39	need	419	2	dad, mom
40	never	480	2	dad, mom
41	nice	653	2	dad, mom
42	night	443	2	dad, mom
43	of	95	2	dad, mom
44	oh	1293	2	dad, mom
45	ok	3802	2	dad, mom
46	old	306	2	dad, mom
47	one	1029	2	dad, mom

ID	Term	FREQ (CORPUS)	NO OF NODES	NODES
48	phone	355	2	dad, mom
49	probably	222	2	dad, mom
50	r	791	2	dad, mom
51	really	1057	2	dad, mom
52	right	778	2	dad, mom
53	room	153	2	dad, mom
54	say	523	2	dad, mom
55	see	1600	2	dad, mom
56	something	512	2	dad, mom
57	soon	269	2	dad, mom
58	sorry	718	2	dad, mom
59	stay	223	2	dad, mom
60	still	498	2	dad, mom
61	sure	728	2	dad, mom
62	take	584	2	dad, mom
63	talk	802	2	dad, mom
64	talking	327	2	dad, mom
65	tell	895	2	dad, mom
66	thats	901	2	dad, mom
67	think	1308	2	dad, mom
68	time	961	2	dad, mom
69	tonight	337	2	dad, mom
70	u	15302	2	dad, mom
71	ur	2039	2	dad, mom
72	wanna	1084	2	dad, mom
73	want	2937	2	dad, mom
74	well	1602	2	dad, mom
75	work	620	2	dad, mom
76	working	151	2	dad, mom

ID	Term	FREQ (CORPUS)	NO OF NODES	NODES
77	would	2391	2	dad, mom
78	www	403	2	dad, mom
79	yea	886	2	dad, mom
80	yeah	1260	2	dad, mom
81	yes	1969	2	dad, mom
82	you	553	2	dad, mom