Detecting Code Words by Large-Scale Language Models

by

Nidhin Pramod

In partial fulfilment of the requirements for the degree of

MSc in

Advanced Computer Science with Big Data

A logo on a black background

Description automatically generated

Department of

Computer and Information Sciences

August, 2023

**Declaration**

This dissertation is submitted in part fulfilment of the requirements for the degree of MSc in Advanced Computer Science with Big Data of the University of Strathclyde.

I declare that this dissertation embodies the results of my own work and that it has been composed by myself.

Following normal academic conventions, I have made due acknowledgement to the work of others.

I declare that I have sought, and received, ethics approval via the Departmental Ethics Committee as appropriate to my research.

I give permission to the University of Strathclyde, Department of Computer and Information Sciences, to provide copies of the dissertation, at cost, to those who may in the future request a copy of the dissertation for private study or research.

I give permission to the University of Strathclyde, Department of Computer and Information Sciences, to place a copy of the dissertation in a publicly available archive.

Yes [ ✓ ] No [ ]

I declare that the word count for this dissertation (excluding the title page, declaration, abstract, acknowledgements, table of contents, list of illustrations, references and appendices is XXXX.

I confirm that I wish this to be assessed as a Type 1 2 3 4 5 Dissertation.

Signature: A signature of a person

Description automatically generated

Date: 12-Aug-2023

Abstract

Acknowledgements

I would like to thank my supervisor Dr Dmitri Roussinov for his guidance and insightful feedback that provided the foundation for this work and helped me steer my research in the right direction. I would also like to thank my parents for their constant belief in my abilities and for supporting me emotionally during times of distress.

Contents

[1 Introduction 1](#_Toc142255296)

[1.1 Problem Definition 3](#_Toc142255297)

[1.2 Research Objectives 3](#_Toc142255298)

[1.3 Applications 4](#_Toc142255299)

[1.4 Contributions 5](#_Toc142255300)

[2 Key Literature 5](#_Toc142255301)

[2.1 Preliminary Research 5](#_Toc142255302)

[2.2 Code word detection using Word2Vec word embeddings. 8](#_Toc142255303)

[A Example appendix 10](#_Toc142255304)

[A.1 Appendix subsection 10](#_Toc142255305)

[B Another appendix 10](#_Toc142255306)

[B.1 A subsection 10](#_Toc142255307)

# Introduction

The English dictionary defines a code word as “*a word or phrase that has a special meaning, different from its normal meaning, for the people who have agreed to use it in this way*” (*Code word definition and meaning | Collins English Dictionary*, no date). Throughout history, code words have been used during communication to disguise sensitive information from everyone except the intended recipient. This was done so that if the communication were to be intercepted, then the actual meaning of the message would not be clear unless the meaning of the code word was known. In times of war, armies often used various code words and cryptographic techniques to ensure the secure transmission of war strategies. In recent times, however, with the advent of the internet, code words have been used in conducting criminal activities online. As of 2021, close to 5 bn people around the world had access to the internet (Ani Petrosyan, 2023) and around 4.6 bn of them have a social media account (S. Dixon, 2023) This has resulted in an increasing number of criminals using social media and dark web to indulge in their illicit activities and expand their network (Moore, 2022). These activities include drug trafficking, planning terrorist attacks, spreading hate speech, etc. (Tom Allard, 2020). To avoid detection from law enforcement agencies and social media content moderators, these criminals often use code words and/or euphemisms to communicate among themselves and their followers or subscribers. Although encrypting messages might sound like the obvious choice for communicating malicious information, it is only used for communication between closed groups of criminals and their associates. However, to convey their messages to a larger audience, most of whom are potential followers or prospective customers, the criminals resort to using code words on social media and darknet platforms. For example, Twitter was used by several terrorist organizations for their recruitment, and it was also popular among wildlife traffickers and drug dealers (*The Crime and Terror Threat on Social Media — ACCO*, no date), (Tassone *et al.*, 2020). In 2018 Reddit banned subreddits that were dedicated to dark-net forums where users frequently indulged in discussion and business of drugs and illegal weapons (Catalin Cimpanu, 2018). Below is an illustration of example sentences containing code words for drugs and weapons:

|  |  |
| --- | --- |
| **Sentences** | **Code word meaning** |
| all up my nose already haha, was a very very fine **crystal** | methamphetamine |
| all his **fish scale** listings are showing as out of stock. | cocaine |
| I was looking to buy some **gats** from the dark web, but the they are out of stock | pistol |

Table ‑ Examples of sentences containing code words

In the Table 1‑1, the words highlighted in block characters in the sentences are the code words. Such code words are used extensively for unlawful activity and detection of such code words is essential in fighting crimes and also mitigating future crimes by curtailing the outreach of criminals. The detection of code words and euphemisms has been studied extensively in linguistics with some studies that date as far back as 1997 (Pfaff, Gibbs and Johnson, 1997). More recently, the problem of code word detection was studied as a Natural Language Processing (NLP) problem using machine learning and deep learning techniques. Natural Language Processing is a branch of computer science and artificial intelligence that helps computers to comprehend and generate human language (Foote, 2023). For many years since its inception, NLP relied on a set of complex handwritten rules and eventually became unpopular due to its ineffectiveness. However, there was a resurgence in this field with the advent of machine-learning algorithms and statistical models in the 1980s and 1990s (Foote, 2023). In recent years, NLP has made remarkable progress with the advent of transformer-based Large Language Models. A Large Language Model is a language model developed using deep neural networks that are trained on large volumes of unlabelled text data, which results in models learning billions of parameters. Parameters here refer to weights and biases learned by the model during training (Zhao *et al.*, 2023).

In this paper, I have studied the code word detection capabilities of Large Language Models (LLM).

## Problem Definition

Detection of code words poses several challenges. To begin with, Human analysts manually analysing posts on social media sites is a monumental task since Twitter alone has 368 million daily active users (*500+ Social Media Statistics You Must Know in 2023*, 2023). Understanding this challenge, social media sites have an option which allows users to report posts that violate the platform’s policies. But due to the presence of code words, the true malicious nature of such posts goes unnoticed and eventually, they don’t get reported as well. With such an approach, real-time code word detection in sentences becomes impossible.

Another challenge is the constantly evolving nature of code words. Existing tools used to detect such code words are inefficient as they rely on a known “ban list” of words, and criminals often switch to using new code words to circumvent moderation policies (Cambridge Consultants, 2019). A study showed that a machine learning-based tool called Perspective meant for the detection of toxic comments could be deceived by modifying the phrases, in other words, by the usage of code words (Hosseini *et al.*, 2017).

Several other machine learning and deep learning models have been proposed for code word detection but many of these models require huge labelled datasets for training, some of them require extensive data preprocessing and involve complex techniques for the detection and identification of code words.

## Research Objectives

The main aim of this paper is to study the code word detection and identification capabilities of large language models such as GPT-3, PaLM 2, facebook/bart-large-mnli and MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli. Through this study, I want to prove that Large Language Models can be used to effectively detect code words and identify what the code word means in real-time, without having to do extensive data preprocessing and without the need for a large dataset to train a model. I want to demonstrate the in-context learning capabilities of Large Language Models, that completely eliminate the need for task-specific training of the model using task-specific datasets (Zhao *et al.*, 2023)

For testing the models, I will be using a real-world dataset that contains sentences having code words in them. Using this dataset, I first want to identify the most optimal prompting technique for each of the language models listed above, by experimenting with zero-shot, one-shot and few-shot prompts. I want to design optimal prompts following the guidelines of prompt programming (Reynolds and McDonell, 2021) and use these prompts to instruct the LLMs to detect code words in the test dataset.

Following this, I want to compare the performance of each of the models and analyse if the size of the LLM, (i.e., the number of parameters) has a direct correlation with the performance of the model.

Finally, I also want to compare the performance of each of these models with a baseline model that is trained on the same dataset.

## Applications

The findings of my research have a wide range of applications. As we will see later in subsequent chapters, GPT-3 and PaLM 2 LLMs can both detect a code word in a sentence and also identify what the code word means. This is very useful for social media content moderators and law-enforcement agencies who can use my framework to detect malicious posts containing code words for criminal substances in real time. This could then be used to mitigate criminal activity on their platforms.

Other smaller LLMs like facebook/bart-large-mnli and MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli can only indicate that a sentence contains a code word. This can also be useful to flag certain posts which are suspected to contain code words for criminal substances, in real time. These flagged posts could then be evaluated through human intervention for further action. These smaller LLMs are open-source models and lightweight which could be used by researchers and smaller organizations for real-time monitoring and code word detection, without the hassle of having to invest in a huge number of resources for the collection and training of data.

## Contributions

My contribution to this research is as follows:

* Preparation of Dataset using BigQuery to extract comments from banned subreddit threads of Reddit.
* Determination of optimal prompts, testing and evaluation of the GPT-3 model.
* Determination of optimal prompts, testing and evaluation of the PaLM 2 model.
* Determination of optimal prompts, testing and evaluation of facebook/bart-large-mnli and MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli models.
* Data preprocessing, training, and testing of FastText classifier.

# Key Literature

This section will contain a review of previous research that has been conducted on code word detection over the years, the current state-of-the-art and recent developments in the field of Natural language processing and Large Language Models which are the core topics of my research. It will also present an overview of key properties of Large Language Models and will also describe briefly the rationale of the FastText classifier.

## Preliminary Research

Skillicorn D (2005) proposed a technique to detect code words used by illicit groups to bypass keyword filtering tools. Keyword filtering is a mechanism that was used to scan intercepted messages for specific keywords that were malicious in nature. The study made use of word frequencies and matrix decomposition techniques to detect anomalous word usage. The rationale behind this approach was the idea that the distribution of words in sentences can be used as a metric to compare two different sentences. Consequently, it would be possible to detect sentences that deviate from expected word frequency distribution, by analysing the decomposed matrices that are generated by applying matrix decomposition techniques such as Singular Value Decomposition and Independent Component Analysis which are described in the paper (Skillicorn, 2005)

Another study also proved that word frequency information can be employed at detection of word substitution or code words in sentences, where a word with similar word frequency is used to replace another word in the sentence (Fong, Roussinov and Skillicorn, 2008). The rationale was that if two words with similar word frequencies, but very different sematic information is used to replace one another, then it creates an anomaly in the sentence that can be detected using a set of measures such as Sentence Oddity, K-gram frequencies, Pointwise Mutual Information, and Hypernym Oddity. Some of these measures treat sentences as a Bag of Words. The Bag of Words is a text representation technique used in Natural Language Processing which represents the frequency in which a word occurs in a given sentence or document (Brownlee, 2017). Sentence Oddity measures the frequency of a Bag of words for a sentence with a target word removed with respect to the frequency of the entire Bag of Words. K-Gram frequency refers to the frequencies of the ‘K’ number of words in a sentence where K is a numeric integer, for example, 1-gram, 2-gram and 3-gram frequencies can be measured. All these measures were then used to identify semantic differences between regular sentences and sentences containing substitutions, to identify code words. The frequency distribution of regular sentences was obtained from Yahoo! Web search interface (Fong, Roussinov and Skillicorn, 2008). A performance analysis performed on these measures in the study by Deshmukh, Deshmukh and Deshmukh (2014) reveals that when these measures were used in tandem with a random forest classifier, the model was effective in detecting code words. There was another technique which was extensively used to represent words for NLP problems during the early 2010s, that is TF-IDF, which stands for Term Frequency and Inverse Document Frequency. Such numerical representations of words in sentences are called word embeddings. The Term Frequency measures the number of times a term appears in a document with respect to the total words in a document, while Inverse Document Frequency measures the rarity of the term in a given document. The formula for Term Frequency is as follows:

A black text on a white background

Description automatically generated

Figure ‑ Term Frequency Formula (Karabiber, no date)

The formula for Inverse Document Frequency is as follows:

A close-up of a text

Description automatically generated

Figure ‑ Inverse Document Frequency Formula (Karabiber, no date)

The final formula for TF-IDF word embedding technique is as follows:

A black text on a white background

Description automatically generated

Figure ‑ TF-IDF Formula(Karabiber, no date)

Both Bag of Words and TF-IDF word embedding techniques have several disadvantages, the major one being that they do not value the order of sentences and do not take into account the semantic meaning of the word in the sentences in which they appear (*Word embeddings in NLP: A Complete Guide*, no date). Due to these reasons such word embeddings were found to be less effective for tasks where understanding the semantic meaning of words in sentences was essential. This problem was solved with the innovation of Word2Vec embedding technique (Mikolov *et al.*, 2013).

## Code word detection using Word2Vec word embeddings.

References

1. Example appendix
   1. Appendix subsection
      1. A subsubsection
2. Another appendix
   1. A subsection
      1. A subsubsection