Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Private Search

Investigating users' agency over search engine learning

Author
Kris Vanhoutte

Supervisor
Prof. Doug Leith

M.A.I. Research Dissertation

Submitted to the University of Dublin, Trinity College, May, 2017

Declaration

I, Kris Vanhoutte, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request.

Signature Date

Summary

This project investigates the effectiveness with which a browser plug-in is able to detect, assess and defend against search-engine learning. Previous research has shown that detection, assessment and defence were all possible with the Google search engine in a scripted environment using the techniques outlined within this paper.

As part of this project, an existing plug-in is further developed to provide robust and reliable operation. This is achieved through an analysis of the HTML events which are triggered by the content delivered by Google, along with extensive unit testing. A combination of foreground and background processes within the plug-in allow it to reliably detect when user searches are carried out, and when webpages, including all internal elements, are fully loaded, all without impacting on the user experience.

The accuracy with which the plug-in is capable of categorising advertisements is greatly improved through the creation of new training data and updating the techniques used. The new training data is evaluated against the existing training data by comparing the rate at which each correctly labels ads. The new training data greatly outperforms the existing training data in the case of all but one category.

One of the core assumptions upon which this project is based is that a commercial search engine will display adverts in order to maximise its expected revenue, and thus evidence of personalisation will be detectable by investigating the changes in adverts shown in response to a neutral probing query. This assumption is evaluated by analysing ad response volume, probe query selection and calculating a score for user-profiling across a range of topics.

The core findings of this project are that the results of the previous research by Mac Aonghusa et al. [4] no longer hold. Results in this paper suggest that Google’s ad policy has changed since the time at which the prior research was conducted. Ad response as a whole is lower, and therefore the assumptions made in [4] do not hold in general. Results outlined

in this paper indicate that these assumptions may hold in certain circumstances, although further testing is necessary to confirm this.

Consequently, the chosen obfuscation method, proxy topic injection, could not be tested, though functionality for carrying out proxy topic searches is included in the plug-in. This functionality differs slightly from that present in [4] as it is necessary to account for the unpredictable nature of user interactions in a non-scripted environment. It is expected that the implementation in this project would provide similar levels of obfuscation to those in [4] if profiling were capable of being assessed.

Overall, the project has not fully achieved its objectives, which were based on assumptions made by the previous research. However, the application created for this project is fully functioning, and could be used as the foundation for future work. The results noted in this work provide valuable insight into the limitations of the previous work, as well as deliver a solid basis for any further work in this field.

Acknowledgements

I would like to thank my supervisor Douglas Leith for providing me with the opportunity to work on this project, meeting with me throughout the project's duration and offering ideas on how best to proceed with the research.

I would also like to thank Pól Mac Aonghusa for his help in brainstorming possible solutions when unanticipated results were reached, as well as taking the time to re-run his tests to confirm we weren't just missing something obvious.

Finally, thank you to all of those who took the time to proof-read this document and provide feedback; Muireann, Tom, Michael and Deirdre.

Contents

Chapter 1

Introduction

This chapter provides an introduction and high level description of the background to the project. It also details the objectives of the project, and the challenges associated with measuring and interfering with the learning carried out by search engines. Finally, it outlines the structure of the remainder of the report.

1.1 Motivation

User-profiling and online personalisation, at their core, involve gathering data about users during their interactions with an online service. This is often done under the guise of providing a more personal and efficient experience to the user. Recommender systems, particularly search engines, are an example of a service which rely heavily on user-profiling. Apart from technical information such as the browser, IP address and operating system used; the data gathered on users can vary from the set of words which users query, to personally identifying information which is requested by the particular service, and everything in between.

The personal data which websites collect on their users is often used for more than just improving the user's experience. While many of these uses may be harmless and go relatively unnoticed by the user, there have been a significant number of cases of this data being abused to make it clear that users should be given more control over what data these companies are collecting and how this data is used.

One of the natural consequences of personalisation, particularly within search engines and social media, is a form of censorship termed a filter bubble in [0]. Filter bubbles occur when users are increasingly provided with results that the service considers potentially “desirable” to them, and so they are exposed less and less to views that conflict with their own opinions, limiting their access to unbiased and contrary results.

In [1], discrimination associated with personalisation is investigated through a review of adverts from Google and Reuters.com. This investigation suggested a strong correlation between a user’s perceived ethnicity based on their name, and adverts suggestive of an arrest record.

In 2006, AOL published a subset of search queries which they had collected over a three month period. They believed that the data was sufficiently anonymised so that no user would be identifiable. However, it was not long before the identity of the first user was uncovered and subsequently the searches, which this user had considered to be private information, are publicly available for the world to see [2]. There are numerous ways in which this sort of data could potentially effect someone’s life, from a negative social impact, to influencing job opportunities and insurance policies.

There are questions about the ethics, or lack thereof, surrounding behavioural advertising online. Companies who use behavioural or targeted advertising dismiss the idea of “opt-in” strategies as unrealistic, citing that without the revenue from advertising they would not be able to operate as the “free” services they provide. However, one must question if a service is truly free if the data collected comes at the cost of privacy and security to their users?

While some search engines and recommender systems allow users to “opt-out” of tracking, this is often difficult for users to achieve, and can become temporary if the user installs a new browser, clears their cookies, etc. Furthermore, certain technical data will always be gathered and search engines can make assumptions based on data with similar technical data. For example, an IP address can be linked to a location and ads provided based searches by users in the same location. The General Data Protection Regulation (GDPR), discussed in more detail in Section 2.2.3, seeks to both provide users with more information about how and why their data is being used, and afford them more control over whether their data is collected in the first place.

Despite regulations such as this, it is likely that companies such as Google and Facebook, who make a large portion of their revenue as a direct result of gathering large amounts of user data, will continue to try and accumulate as much of that data as possible. As such, there will be an ongoing demand for applications that provide users with ways of defending themselves against this data collection.

1.2 Objectives

The objective of this project is to create a plug-in for use in commercial web browsers that allows users to measure and control search engine learning with regards to a set of sensitive topics. Search engine learning, as used here, refers to a search engine perceiving a user to have significant interest in a particular topic or set of topics. To achieve this objective a number of distinct subtasks were identified:

1. Carry out comprehensive unit-testing of the existing plug-in to identify and fix issues, as well as provide a framework for unit-testing for any further development.
2. Further develop the plug-in to use the **PRI+** measure and allow for proxy topic obfuscation as discussed in [4].
3. Develop a testing framework to evaluate the plug-in against results obtained in prior work and existing plug-ins which provide similar functionality.

Based on the objective and subtasks above, measures of success for the project were established.

1. Obtain robust and reliable functionality of the plug-in.
2. Achieve accurate functionality of the plug-in using the **PRI+** measure.
3. Comprehensively evaluate the plug-in's ability to detect and assess search engine learning.
4. Disrupt search engine learning through the use of proxy topic obfuscation.

1.3 Challenges

When working with any sort of online system, particularly ones in which information related to the inner functionality of that system is limited, the challenges faced are often much broader than simple programming challenges. These challenges only increase when the online system in question is as large and complex as Google's search engine. Some of the challenges that were faced within the project are detailed below.

- Event recognition: In order to optimize user experience, Google Search results are loaded asynchronously and various page elements are loaded and applied after the page load event has fired. For example, the advert style is only applied after the page content has been displayed. On top of this, with subsequent searches, the page is updated rather than reloaded. The perceived response time is further improved by the page scripts executing searches as the user is typing a new query. Finally, Google uses the auto complete text in the search box to try and pre-empt searches.

The net effect of all of these factors is that one cannot rely solely on standard HTML events to determine when a page has been fully loaded. Instead, an event listener is required, which checks for page load events and performs certain content checks to confirm that a new search is complete. A timing element for repeat checks is also required to account for intermittent searches as a user types in a new query.

- Google's security measures: Google has measures in place to detect and combat many different types of cyber-activity which they consider to be malicious. These include cyber-attacks such as DDOS (Distributed Denial of Service) attacks, and automated searching by bots. These security measures proved to be an obstacle to testing features of the plug-in, especially when trying to find the most effective way to introduce proxy topic obfuscation.

- Incorrect assumptions: Google Search, like many online systems, does not provide source code or extensive details about the inner-workings of, and algorithms behind, their systems. Consequently, when trying to disrupt one of these systems, many assumptions need to be made regarding the unknown mechanisms in order to simplify the problem. Unfortunately, over simplification and incorrect assumptions can lead to difficulty obtaining meaningful results.

  Based on the assumption that search engine learning could be detected and assessed by looking at the ad response to probe queries, it was expected that results observed in this project would correspond to those in [4]. When no personalisation was discernible, a re-assessment of the previous assumptions and results became necessary.

- Browser behaviour: The previous research, which this project is based on, conducted tests using a headless browser [31] in a scripted environment. The Chrome WebDriver was used for this project, the key difference being that it is GUI based. Not all scenarios can be covered with a purely scripted browser, and several of these proved disruptive to automated testing in this project, such as the appearance of pop-ups and redirects when a click strategy was used.

Some of the technical challenges which were encountered during this project are listed below:

- Working from an existing code-base, building on it and determining the flaws that exist within it.
- Gaining familiarity with a new programming language, JavaScript, and learning to carry out comprehensive unit testing with QUnit.
- Language constraints; lack of support for multithreading in JavaScript, and recent deprecation of certain functionality related to synchronicity in the background threads.

1.4 Outline

Chapter 2 of this report provides further background information on this project. It explains the techniques used to detect and assess learning by search engines, and the methods used to disrupt this learning. Furthermore, it introduces some examples of related research, practical work and social policy, and explains the reasoning for the selection of technologies used.

Chapter 3 details the design and implementation of the software side of the plug-in. First it provides an overview of the functionality of the plug-in, followed by a discussion of the development work carried out on the plug-in.

Chapter 4 describes the methods which were used to evaluate the plug-in and the results which were obtained following this evaluation. It concludes with an overall discussion of the results which were obtained within the context of the objectives outlined in this chapter.

Chapter 5 completes the report with a set of conclusions which have been reached based on the results obtained throughout the project's development. Potential for future work which could be conducted is also discussed.

Chapter 2

Background and Related Work

This chapter begins with a discussion the ways in which search engines such as Google gather data on their users and provide personalisation. It then explains the techniques that are used for assessing the learning done by the search engine, and the techniques which can be used to defend against this learning. Examples of related research, practical examples and social policy are provided. Finally, the choice of programming languages, supporting libraries, and web browsers is discussed.

2.1 Background

Firstly, this section examines how search engine personalisation works on a broad level, and then notes the assumptions about the Google search engine that are made as part of this project. It then outlines the method used to assess learning, first in general terms, and then through the introduction of notation and formulae. Finally, it briefly discusses a number of methods used for defending against search engine learning, with a more detailed discussion of proxy topic injection, which is the obfuscation method used within this project.

2.1.1 User-Profiling and Search Engine Personalisation

User-profiling occurs across a vast portion of the internet, through the use of user accounts, monitoring of user activity with cookies, and various other methods. Some of the most comprehensive internet user-profiling which exists today is carried out by companies such as Facebook and Google. The services these companies provide are integrated into many other websites across the web, such as Google AdSense and Facebook’s ‘Like’ and ‘Share’ buttons. Not only do these companies track users on their own domains, they also utilise the integration of their services to track users across domains they do not own.

Websites utilise many different web technologies to carry out this tracking, from storing data on the user's computer using cookies, to executing tracking programs through the use of JavaScript. Attempting to disrupt all forms of user-profiling would be a monumental task, and would likely have a significant impact on a user's experience, as many of the technologies used to track are also used to provide features for users. In this project the focus is solely on the user-profiling which is carried out by search engines on their own domains, with a particular focus on the Google search engine.

It is recognised that the search engine personalisation which is visible on *google.com* and associated domains is likely due to user-profiling on a broader scale than simply the interaction which users have with Google's search engine. However, it is hoped that despite this, evidence of learning can be detected by looking at changes to personalisation in response to interaction solely with the search engine itself. To help ensure this, all browsing data is cleared between tests, and the web browsers in use are utilised exclusively for this project, with no personal browsing occurring.

Several different assumptions are made in this project, which are detailed below, regarding the ways in which a search engine learns of its users' interests. These assumptions are made with the intent of simplifying the problem of assessing the user-profiling being carried out by the search engine. The search engine is modelled as a black-box and minimal assumptions are made about the internal details of the algorithms, as was done in [3, 4].

Assumptions

Assumption 1**:** Search engine learning primarily takes a user's search query as input, along with any interaction a user has with the results page, i.e. clicking of ads and/or links.

Assumption 2**:** A for-profit commercial recommender system, such as a search engine, selects variable page content to maximise its expected revenue. In particular, when a search engine infers that a particular advertising category is likely to be of interest to a user, and it is more likely to generate click through and sales, it is obliged to use this information when selecting which adverts to display. This suggests that, by examining advert content recommended by the search engine, it is possible to detect evidence of sensitive topic profiling by the search engine.

Assumption 3**:** The background knowledge at the first step of a query session, $\mathcal{E}_1$, provides sufficient description of background knowledge for all subsequent steps of that query session, $\mathcal{E}_k$.

Assumption 4**:** Adverts are selected to reflect search engine belief in user interests. Thus adverts are assumed to be the principal way in which search engine learning is revealed.

2.1.2 Assessment Model

In contrast to a proportion of other research in this area [6, 7], this project utilises its own method of assessing learning by a search engine, rather than relying on the information provided by Google. Google has a set list of potential topics with which it categorises user interests, and while these topics will broadly cover most user searches, they do not allow for the specific detection of interest into what might be considered ‘sensitive’ topics, such as sexuality or unemployment.

This work attempts to detect and measure learning through the use of a Privacy for Individuals (**PRI**) value, which was first proposed in [3], and later further improved in [4], where the notation was changed to **PRI+**. The sensitive topics of interest in this project are the same ones used in [3, 4].

Assumption 2 above is the key assumption behind the detection and assessment model proposed in [3, 4], and utilised in this project. In order to test for learning, a predefined probe query is injected into a stream of user queries during a query session. In this way, any differences detected in advert content in response to probe queries can be investigated to identify evidence of learning. Effort is made to minimise bias in the probe queries and there is no user interaction with the response page for probe queries, thus minimalizing any contribution to search engine learning by the probe query.

Assumption 3 implies that it is not necessary to use knowledge of the search history during the current session when estimating the **PRI** score for a topic c as this is already reflected in the search engine response and the background knowledge $\mathcal{E}_1$ at interaction k. This assumption greatly simplifies estimation as it means it is not necessary to take account of the full search history, but requires that the response to a query reveals search engine learning of interest in sensitive category c which has occurred. This assumption was called the "Informative Probe" assumption in [3] and the "Sufficiently Informative Response" assumption in [4].

Notation and Formulae

Let $\{c_1, \dots , c_N\}$ denote a set of sensitive categories of interest to an individual user, e.g. sexuality, cancer, unemployment, etc. Gather all other uninteresting categories into a catch-all category, 'non-sensitive', denoted c_0. The set $C = \{c_0, c_1, \dots , c_N\}$ is *complete* in the sense that all user topic interests can be represented as subsets of C with the usual set operations.

It is assumed that a user interacts with a search engine by issuing a query, receiving a web page in response and then clicking on one or more items in the response. A single such interaction, labelled with index j, consists of a *query*, *response page*, *item-click* triple, denoted $\Omega_j = (q_j, p_j, l_j)$. A user session of length $k > 0$ interactions consists of a sequence of k individual interactions, and is denoted $\{\Omega_k\}_{k\geq1}$. The sequence of interactions $\{\Omega_k\}_{k\geq1}$ is jointly observed by the user and the search engine – and perhaps several other third-party observers.

From ***Assumption 3***, the background knowledge $\mathcal{E}_1$ is estimated by selecting a training dataset, denoted $\mathcal{T}$, consisting of (label, advert) pairs; where the label is the category in C associated with the corresponding advert. For example, when testing for evidence of a single sensitive topic, called 'sensitive', $\mathcal{T}$ contains items labelled 'sensitive' or 'non-sensitive', where 'non-sensitive' is the label for the uninteresting, catch-all topic c_0. In this way $\mathcal{T}$ approximates the prior observation evidence available at the start of the query session so that $\mathcal{T}$ is an estimator for $\mathcal{E}_1$.

Text processing of $\mathcal{T}$ produces a dictionary $\mathcal{D}$ of keywords. This processing removes common English language high-frequency words and maps each of the remaining keywords to a stemmed form by removing standard prefixes and suffixes such as "-ing" and "-ed". The dictionary $\mathcal{D}$ represents an estimate of the known universe of keywords according to the background knowledge contained in the training data.

Text appearing in the adverts in a response page is pre-processed in the same way as $\mathcal{T}$ to produce a sequence of keywords from $\mathcal{D}$ for each advert; denoted $W = \{w_1, w_2, \dots , w_{|W|}\}$. Let $n_{\mathcal{D}}(x|W) := |\{i : i \in \{1, \dots , |W|\}, w_i = x\}|$, denote the number of times with which an individual keyword $x \in D$ occurs in a sequence $W = \{w_1, w_2, \dots, w_{|W|}\}$. The relative frequency of an individual keyword $w \in W$ is therefore[1],

$$\phi_{\mathcal{D}}(w|W) = \frac{n_{\mathcal{D}}(w|W)}{\sum_{d \in \mathcal{D}} n_{\mathcal{D}}(d|W)}$$

recalling that only keywords w appearing in $\mathcal{D}$ are admissible due to the text pre-processing.

In the original **PRI** estimator, word frequency given adverts a_k on the result page for query number k was calculated using $\phi_{\mathcal{D}}(w|a_k)$. This proves to be problematic when the adverts a_k do not contain any of the topic keywords in the dictionary $\mathcal{D}$, i.e. when $a_k = \emptyset$. Such adverts indicate that there is no detectable evidence of interest in a particular topic, and so to be consistent with the definition of ε-*Indistinguishability* in Section 2.3 of [4], $\phi_{\mathcal{D}}(w|a_k)$ was replaced with

$$\varphi_{0,\mathcal{D}}(w|a_k) = \begin{cases} \phi_{\mathcal{D}}(w|a_k) & if a_k \neq \emptyset \\ 1 & otherwise \end{cases}$$

A regularisation approach was also adopted in [4], to account for the fact that the training data is based on a limited sample of adverts and that it is possible for the training data to observe no adverts containing an infrequently occurring keyword. The following equations were thus defined.

$$n_{\lambda,\mathcal{D}}(w|W) = \lambda + (1 - \lambda) n_{\mathcal{D}}(w|W)$$

$$\phi_{\lambda,\mathcal{D}}(w|W) = \frac{n_{\lambda,\mathcal{D}}(w|W)}{\sum_{d \in \mathcal{D}} n_{\lambda,\mathcal{D}}(d|W)}$$

$$\varphi_{\lambda,\mathcal{D}}(w|a_k) = \begin{cases} \phi_{\lambda,\mathcal{D}}(w|a_k) & if a_k \neq \emptyset \\ 1 & otherwise \end{cases}$$

The parameter $0 \leq \lambda < 1$ enforces a minimum frequency of $1/|\mathcal{D}|$ on every keyword.

Let $c_i \in C$ be a sensitive topic of interest, and let $\mathcal{T}(c_i)$ denote the subset of $\mathcal{T}$ where the labels correspond to c_i. Let $\mathcal{T}(C)$ denote the set of adverts labelled for any topic in C. Finally let a_k be the adverts appearing on the result page for query number k. The **PRI+** estimator is thus given by the equation

$$\widehat{\mathbb{M}}_k(\bar{c}_i, \Omega_k) = \sum_{w \in \mathcal{D}} \left(\frac{\phi_{\lambda,\mathcal{D}}(w|\mathcal{T}(c_i))}{\phi_{\lambda,\mathcal{D}}(w|\mathcal{T}(C))} \cdot \varphi_{\lambda,\mathcal{D}}(w|a_k) \right)$$

[1] The notation used here differs from that in [3, 4]. These changes to notation were made to reduce ambiguity and re-use of similar variables.

The output from **PRI+**, is an estimate $\widehat{\mathbb{M}}_k(\bar{c}_i, \Omega_k)$ for each topic category $i = 0, \dots, N$. These estimates are gathered into an $N + 1$ dimensional vector $\overline{\boldsymbol{P}}$ which is called the **PRI+** score, so that the *ith* component $\mathrm{P_i}$ of the **PRI+** score is $\widehat{\mathbb{M}}_k(\bar{c}_i, \Omega_k)$. This **PRI+** score was normalised in [4], but this normalisation was not included in this project, as is discussed in Section 3.5.

In [4], the value of the regularisation parameter λ was chosen so the prior probability distribution is approximately equiprobable by minimising the square error loss function, i.e.

$$\lambda \in \arg\min_{\lambda \geq 0} L(\lambda) := \sum_{i=0}^{N} \left(\frac{1}{N+1} - \hat{p}_{\lambda,i} \right)^2$$

where

$$\hat{p}_{\lambda,i} := \sum_{w \in \mathcal{D}} \hat{f}_{\lambda,i}(w)$$

$$\hat{f}_{\lambda,i}(w) := \frac{n_{\lambda,\mathcal{D}}\big(w \big| \mathcal{T}(c_i)\big)}{\sum_{d \in \mathcal{D}} n_{\lambda,\mathcal{D}}\big(d \big| \mathcal{T}(C)\big)}$$

Upon investigation as part of this project, it was discovered that these equations have a trivial solution of $\lambda \approx 1$. Calculating a new set of equations to select λ, such that the prior probability distribution is approximately equiprobable, was not possible within the time frame of this project, and so a constant value of $\lambda = 0.01$ was used throughout.

2.1.3 Defence Model

In [4], three distinct defence models were investigated. The most effective of the three was *Proxy Topic Injection*, which is what is used within this project, and shall be discussed in more detail shortly. First, the other two methods which were investigated are briefly described, as well as the results observed for those methods.

The *Random Noise* model involved selecting *noise queries* randomly from a list of popular queries and injecting them into a sensitive query session. The idea here is that such randomly selected queries emulate user interest in the catch-all 'non-sensitive' topic, c_0. Three different ratios of noise to sensitive queries were used; 1:1, 2:1 and 3:1, although none of these provided consistent reduction in the rate at which user interest in a particular topic was determined.

The *Click Strategies* model involved four different click strategies, which were compared to a *no click* baseline test. The click strategies tested were *click relevant*, *click non-relevant*, *click all* and *click 2 random items*. Relevance of links and ads on the results page was calculated based on term-frequency of the words present and compared to a constant threshold value. The *click all* method provided the best results, with an average reduction of 7% for the correct detection of interest in a specific topic.

Proxy Topic Injection

The *Proxy Topic Injection* strategy is similar to the *Random Noise* model, but rather than injecting random queries into a stream of sensitive user queries, a sequence of queries, all related to the same uninteresting proxy topic, are injected into a user session. This sequence of queries seeks to exploit the results observed in the *Random Noise* tests; that is, isolated queries appear not to tend to provoke search engine learning. So, by interspersing the sensitive user queries amidst a stream of uninteresting proxy topics, the sensitive user queries might appear as random noise and not provoke search engine learning, while the uninteresting proxy topics will appear as the users true interest.

In order to test the effectiveness of this model in [4], probe queries were placed before and after each block of 3-4 proxy + 1 sensitive queries to measure changes in **PRI+** score. A session consisted of a total of 5 probe queries, with the order of proxy and sensitive queries being randomly shuffled between each probe. An example of a proxy topic injection session is:

probe, proxy, sensitive, proxy, proxy, probe, proxy, proxy, sensitive, proxy, proxy, probe, sensitive, proxy, proxy, proxy, probe, proxy, proxy, proxy, sensitive, probe

The results of using proxy topic injection were that the True Positive detection rate for all topics and all click-models tested was 0%. The authors of [4] find no evidence of learning by the search engine when proxy topic injection is included. Their previous tests prove that their detection method is notably sensitive to search engine learning, and they can thus conclude that the results they obtained over a series of 2,300 sessions reflect successful misdirection of the search engine away from the sensitive topics tested.

2.2 Related Work

There have been many different approaches taken to providing users with greater agency over their personal data and the ability to disrupt search engine learning. These approaches have come from both the research community and the wider internet community, but none have yet provided functionality which gives users full control. It has been shown that it is possible to detect the efforts of many of the existing solutions. In this regard, the search engines and recommender systems which are carrying out user-profiling remain the state of the art in this field.

In the following sections the research literature on the topic is discussed, with comparisons drawn between the methods used. A brief evaluation of some of the plug-ins which are commercially available is provided. Lastly, the General Data Protection Regulation is introduced, explaining the effects it might have on the future of this field.

2.2.1 Related Research

There are an extensive number of different approaches in the research literature taken to both detection of learning and attempted obfuscation. A subset of the most relevant research is described below, with a summary of the methods used and results achieved given, as well as a brief critique on the limitations of the research.

In [5], plausibly deniable search is achieved through the use of a combination of standardising queries and using 'cover' queries. The query generated by the user is never delivered to the search engine. Instead, the user query is substituted for a standard, closely-related query intended to fetch the desired results. In addition, a set of $k-1$ unrelated 'cover' queries are delivered alongside the substitute queries. The system is designed in such a way that any of the k queries will produce the same set of k queries, thus resulting in k-deniability of interest in any particular topic.

This research was operated offline and on a closed dataset, with all of the canonical (user) and cover queries being precomputed. In the conclusion the authors recognise that this is very much preliminary work, and that a dynamic solution would be desirable for future work. This research is also somewhat limited by the fact that the authors know the full functionality of the search engine which they are interacting with, which would not be the case for a live search engine such as Google or Yahoo!.

The information that Google provides on its perceived interests was used to determine learning in [6]. The authors took a set of website links from Reddit and investigated the browsing profiles that were generated by Google when these websites were visited. The profiles created had a heavy bias towards certain interest categories. Using this information, they then generated a set of *anti-profiles*, whereby they compiled a set of links that would create a profile with a high bias towards the previously uninteresting topics. This *anti-profile* can then be used alongside the original profile to obscure the user-profiling being carried out by Google.

The authors show that their method measurably influences the interest profile generated by Google, but acknowledge that even without the use of obfuscation strategies, the interest profile that Google builds can vary greatly over time.

TrackMeNot (TMN) [7, 8] is a plug-in that attempts to obfuscate user searches amidst a stream of programmatically generated decoy searches. TrackMeNot employs a mechanism for creating dynamic query lists, which allows it to avoid being detected by systems that note repetitive queries. It also uses two different query operation modes, one based on random intervals and the other based upon sending bursts of queries around a user search. The ratio of use for these is definable by the user. Finally, TMN aims to minimise its impact on pay-per click advertisements by avoiding clicking on ads that it identifies as potentially revenue-generating.

In [8], the effectiveness of TMN is evaluated by looking at the Yahoo! interest profile that is generated in response to user searches and obfuscation. There are advantages and disadvantages to relying on the interest profile provided by a search engine, a method that is also employed in [6]. The main advantage is that there is a well-defined set of interests with which to test the effectiveness of any employed strategies. Unfortunately, this advantage can also be one of the biggest limitations with any profiling detection strategy, as it relies on the good-will of the service provider in comprehensively and reliably providing information on the categories which they use to profile users.

2.2.2 Existing Plug-Ins

Plug-ins that provide users with more control over the tracking occurring on the internet are hardly new phenomena. Two of the main approaches that are taken in this regard are the use of Do Not Track (DNT) headers and Virtual Private Networks (VPNs), each of which is implemented in many different plug-ins in both the Firefox and Chrome web stores.

Do Not Track is a technology and policy proposal that enables users to opt out of tracking by websites they do not visit, including analytics services, advertising networks, and social platforms. Do Not Track signals a user’s opt-out preference with a HTTP header, a simple technology that is completely compatible with the existing web. While some third parties have committed to honour Do Not Track, many more have not [9]. In [10], it was shown that using DNT headers had very little effect on the number of cookies that were being used to track users. Plug-ins that turn all requests into Do Not Track requests cannot be guaranteed to actually prevent tracking by websites.

A VPN is the extension of a private network that encompasses links across shared or public networks like the internet [11]. In essence, this allows those using the VPN to create a secure and encrypted connection between two end points. VPNs are often used in conjunction with proxy servers to protect personal identity and location. The proxy server acts as an intermediary to the website that the user wants to connect to. The user communicates securely with the proxy server via the VPN, and the proxy server communicates with the end website. Many plug-ins that use VPNs and proxy servers provide users with the option to vary the server being connected to, which can make a single user’s traffic appear to come from different sources with each request to the end website.

While both of these are general solutions to the tracking which occurs on the internet, there are also plug-ins which attempt to specifically disrupt some of the negative effects of search engine learning and optimization, particularly that done by Google. Two of these plug-ins have been discussed in more detail below, although there are many others which have very similar functionality.

Random Walk

One such plug-in is the Random Walk plug-in for Firefox [12]. The aim of this plug-in is to “deliberately degrade the informational 'signal-to-noise' ratio in the activity logs kept by your ISP, search providers, etc.” [12]. This plug-in allows a user to take a ‘random walk’ through the internet by clicking links at random on the current page. It provides manual or automatic modes of operation, and if it reaches a page it has already visited, it starts down a new path by carrying out a random search.

The result of this ‘random walk’ is that users can escape the filter bubbles which are created by search engines and social media sites and encounter “serendipitous discoveries and inspiration”. Unfortunately, while it might provide a brief reprieve from the filter bubble norm, it does very little to combat the learning which occurs during a typical user session.

Removing Google's Redirects

Another plug-in, which attempts to take a more direct approach to disrupting search engine learning, is the Google Redirects Fixer & Tracking Remover plug-in for Firefox [13]. This plug-in works on the assumption that Google learns through the use of a redirect link which tracks the links a user clicks on before sending them on to the desired URL. The plug-in removes the redirection and instead sends clicks to the desired destination directly. There are a number of plug-ins for both Firefox and Chrome which implement the same or very similar functionality, but this plug-in has the largest user base.

While removing the redirect link does prevent Google from being able to track which link was clicked on through their own methods, they have the potential to learn through persistent cookies, as well as tracking functionality which is present on many websites which use Google services. On top of this, research has shown that Google learns more through the searches users carry out, rather than the links that they click on [4].

The fact that these plug-ins have tens of thousands of users makes it clear that there is a demand for services which provide users with more control over their own data and what search engines learn from them.

2.2.3 General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) [14] was passed by the EU Parliament in April 2016. Set for enforcement in May 2018, this policy is likely to cause substantial changes to the ways in which online data is collected and utilised. Some of the key points of the regulation are detailed below.

Data subjects, which here refers to anyone whose personal data is collected by an online organisation in their interaction with that organisation, have the right to access to information regarding their personal data. This includes whether or not their data is being processed, where and for what purpose, as well as a full copy of their personal data.

Privacy by design shall be a legal requirement. The two significant effects of this requirement are that controllers will only be able to hold and process that data which is necessary for the completion of its duties (data minimisation) and will be required to limit access to personal data to those needing to act out the processing.

The conditions for consent have been made stronger and more explicit. Companies are now prohibited from using long illegible terms and conditions to request consent from users. The request for consent must be given in an intelligible and easily accessible form, and the purpose for data processing must be stated clearly.

There are numerous more specifications as part of this regulation, but those listed above are the most relevant to this project. It remains to be seen if noticeable change will appear as a result of these new regulations. It seems likely that large companies, who make billions of dollars each year through collection and processing of user data, will do their best to find loopholes and work around whatever legislation comes into place. Small- and medium-sized businesses will feel the most pressure to quickly fall into line with these regulations, and it is interacting with these organisations that the most visible differences will be seen.

2.3 Development Software

In developing the plug-in for this project, there were an abundance of options for programming languages, supporting libraries and web browsers with which to work. In this section the reasoning behind the decisions which were made are briefly explained.

2.3.1 WebExtensions

As the aim of this project was to develop a plug-in for use in commercial browsers, it was decided to use WebExtensions as the system for development. WebExtensions is a cross-browser system for developing plug-ins. To a large extent the API is compatible with the extension API supported by Google Chrome and Opera. Extensions written for these browsers will in most cases run in Firefox or Microsoft Edge with just a few changes [15]. The plug-in itself was written using standard Web technologies – JavaScript, HTML and CSS – plus some dedicated JavaScript APIs.

2.3.2 QUnit

Much of the data within this project is stored in large, sparse matrices, which are quite difficult to debug by stepping through code line by line. This fact, coupled with the complexity of some of the equations, meant that unit testing was a necessity. The introduction of unit testing allowed for a significant number of minor errors, which had significant consequences, to be discovered.

In deciding on a unit testing framework to use, a few different frameworks were considered. The three main frameworks which were looked at were Mocha [16], Jasmine [17] and QUnit [18]. QUnit was ultimately chosen, as it was the only framework that did not require a download to use, and has a very simple and easy-to-use structure, which made it perfect for the purpose of this project.

2.3.3 Selenium

Selenium is a suite of tools used to automate web browsers across multiple platforms [19]. It is particularly useful for automating the testing of web applications on those browsers. Selenium supports the WebDriver API and its underlying technologies, which describe a language neutral coding interface for browsers [20]. Selenium was used in the scripted testing for [3, 4], and has a large amount of documentation available for testing against Chrome and Firefox, which made it the framework of choice for this project.

2.3.4 Python

Python was selected as the language to use for the scripted testing of the plug-in in this project. This was primarily because the tests which were carried out in [3, 4] used Python, and were made available for use in this project. It therefore made sense to continue using Python to minimise differences between the testing environment in this project and that in [3, 4]. Python 3 was used for all scripted tests that were run as part of this project.

2.3.5 Web Browsers

The plug-in was developed for the Firefox and Chrome web browsers. Together these two web browsers account for between 66% and 76% of the usage share of desktop browsers as of December 2016 [21, 22]. As such, should the plug-in reach a deployable state, it would be possible to make it available to a significant portion of the internet user base.

As mentioned in Section 2.3.3, both Firefox and Chrome have WebDrivers that are compatible with Selenium which is used for scripted testing of the plug-in. Both browsers expose a similar, Chromium based, API, and so the plug-in can be deployed in both browsers. Running test scripts only requires loading a different driver, all other functionality remains the same. Due to the compatibility between the two browsers, only the Chrome WebDriver was used during the tests run for this project.

Chapter 3

Software Design and Implementation

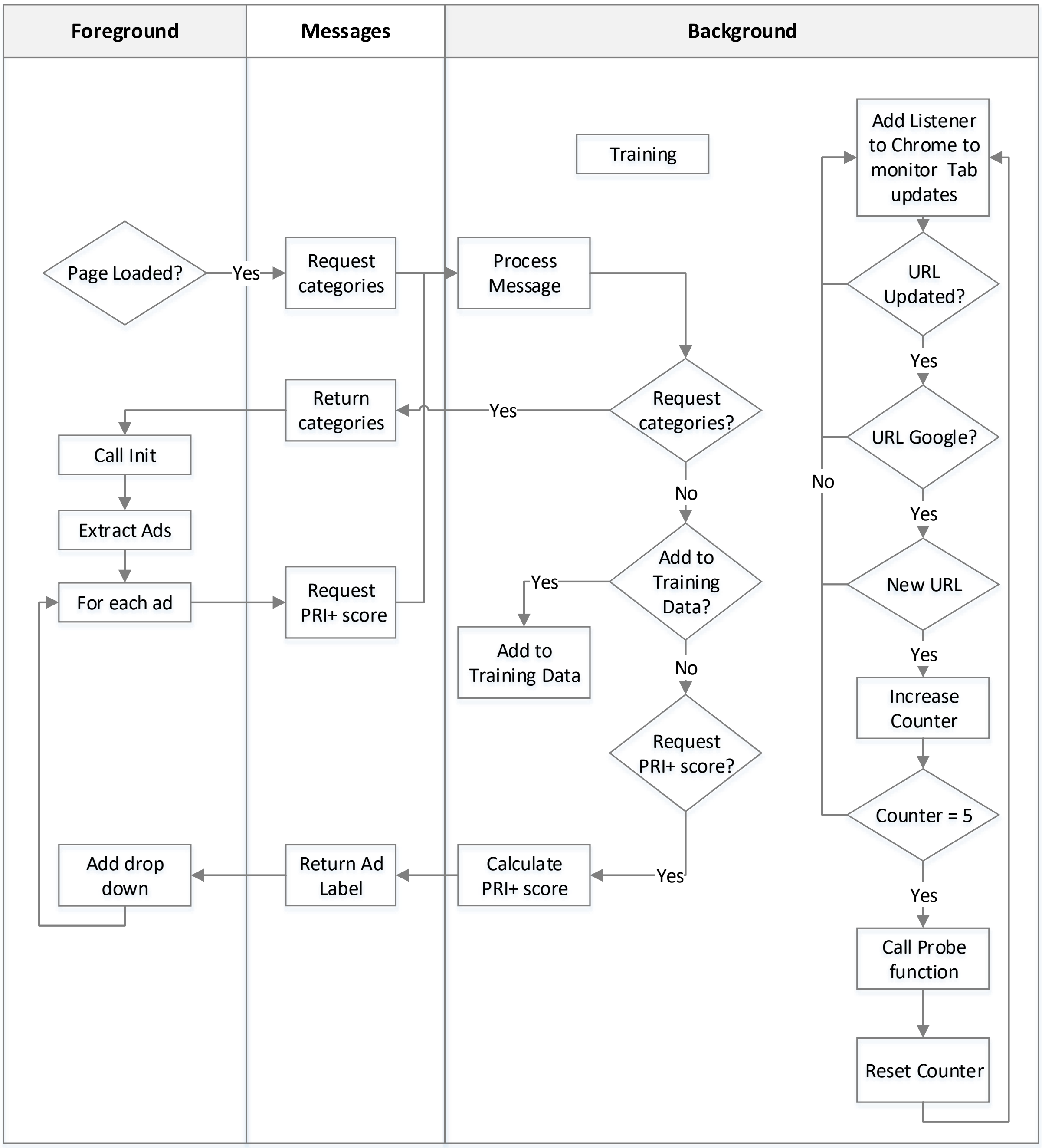
This chapter details the implementation of the project. At the beginning of this project, a pre-existing plug-in was provided which had some functionality required for the application already in place. While this was of huge benefit in getting started, the operation of this plug-in was not reliable, correct, or accurate.

The first section of this chapter provides an overview of the functionality present in the plug-in at the end of this project. In the subsequent sections, details are provided regarding how the functionality of the plug-in was made reliable and robust, correct in terms of the implementation of **PRI/PRI+**, and accurate with respect to the classification of ads in the user and probe pages.

3.1 Overview

The functionality of the plug-in is split between two main processes, as shown in the flow chart below. From a user experience perspective, the impact that the plug-in has on a user's typical browsing session is minimal. The ads that the user sees on *google.com* and associated domains are tagged with a suggested category, and an icon is present at the top of the browser window, which allows users to view a heat map of their current perceived interests. Processing in the foreground, i.e. the user thread, is kept to a minimum to avoid causing lag in the display of pages for the user.

The majority of the processing occurs in the background thread. When the plug-in is initially installed, a training function is carried out which loads in the training data string, parses it, and creates a set of arrays that are stored by the plug-in and used in the calculation of **PRI+** scores. The background thread has two event listeners. The first awaits messages from the foreground, processes them and in some cases provides a response. The second listener is discussed in detail in Section 3.2, but is used to monitor when new searches are carried out and controls the probe queries.

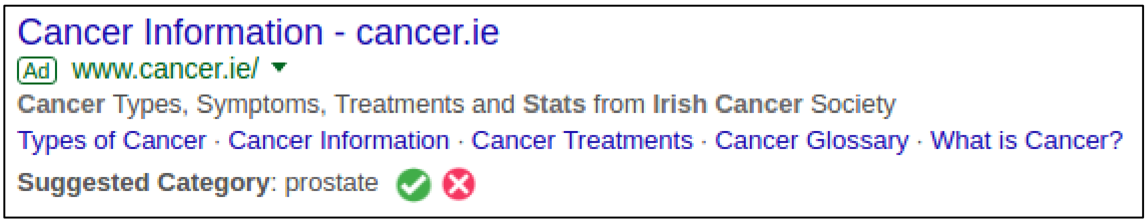


Flow Chart of Foreground and Background processes

It should be noted that the proxy topic injection process has been excluded from the above flow chart. This functionality is implemented in the final version of the plug-in, but because of difficulties in evaluating performance it was decided to omit it from this chart.

3.2 Front-End Integration

One of the desired features of the application is that it should allow the user to add to the training data, to increase accuracy and allow for continual learning over time. This is implemented as follows: each ad shown in response to user queries is extracted from the current page, **PRI+** scores are calculated for it, and a category is assigned to the ad by taking the highest of the **PRI+** scores. This assigned category is then displayed below the advert in the response page as shown below.



Cancer Information - cancer.ie
Ad www.cancer.ie/ ▾
Cancer Types, Symptoms, Treatments and **Stats** from **Irish Cancer** Society
Types of Cancer · Cancer Information · Cancer Treatments · Cancer Glossary · What is Cancer?
Suggested Category: prostate ✓ ✗

Image I: Example of Suggested Category for an Ad

Users have the option to accept the suggested category, in which case the advert text is processed in the same manner as discussed in Section 2.1.2. Any new words that were present in the advert but not in the existing dictionary $\mathcal{D}$ are added to the dictionary.

Detecting page load events and user searches

All the above relies on the ability to detect when the page has been fully loaded, so that the ads that are displayed are available for extraction. For most websites this is a very simple process of using one of the standard events for detecting when a page is loaded. As part of this project, the following events were explored: DOMContentLoaded [23], readystatechange [24], load [25], loadend [26], and progress [27].

In the initial version of the plug-in, an event listener was implemented that waited to receive a DOMContentLoaded event. Unfortunately this method proved to only provide reliable functionality on certain operating systems and browsers. In most cases, the event either fired in advance of the ads being loaded, in which case they were not labelled, or it fired repeatedly, resulting in suggested categories appearing multiple times.

The reason that this and other standard methods for detecting page loading are ineffective is because Google search results are loaded asynchronously in order to optimize user experience. This means that various elements are loaded and applied after the page load event has fired. On top of this, with subsequent searches, the page is updated rather than reloaded. This makes it very difficult to determine when searches are completed with standard HTML events.

The solution within this project is to use polling in the form of an interval function in the foreground to monitor the `readyState` of the document. According to the Mozilla Developer Network, `document.Readystate` returns “loading while the document is still loading, interactive when the document has finished loading and the document has been parsed but sub-resources such as images, stylesheets and frames are still loading, and complete when the document and all sub-resources have finished loading” [28].

While classic polling has its limitations [29], effort is made to minimise the performance impact this polling has by only polling once per second, and minimising the complexity of the polling function. If the page has fully loaded, the URL is checked to see if the current page is a Google domain, and that the ads have not already been labelled. If this check fails, the polling function returns to its waiting state. Otherwise, the ads on the page are processed and categories requested for them.

One key advantage of polling is that it is unaffected by the fact that Google updates the page rather than reloading it when consecutive searches are carried out. The polling function runs continuously until the browser is closed, this means that any news ads will be noticed and labelled when the page is updated and the previous ad categories removed.

On top of recognising when the page is fully loaded, it is necessary to determine when a user search is carried out. The only case of interest is when a new search is carried out, rather than when the current search is returned to after clicking a link or reloading the page. As such the polling function cannot be relied upon to determine new searches. Instead, URL changes are monitored in the background thread according to the following logic.

```
if (tab.OnUpdated event triggered)
    if (URL is a Google URL)
        if (Current URL != Previous URL)
            SearchCount++
            if (SearchCount > 4)
                Probe()
                SearchCount = 0
```

3.3 Background Functionality

Unit-Testing

As mentioned in Section 2.3.2, QUnit was used to unit-test the existing application. Using QUnit it is possible to unit test any function which has a return value, by comparing the return value of the function with an expected return value. The application was split according to the JavaScript files within the project, and for each file, a set of unit-tests were carried out to comprehensively test each function. Comprehensive as used here means that all returning functions were tested, and each conditional branch within these functions was tested.

By testing in this manner, it was possible to discover a significant number of errors within the code. These errors ranged from minor mathematical errors to indexing errors that caused the **PRI** scores to be incorrectly stored. After fixing these errors and retesting to ensure correct functionality, it allowed further development to be carried out in the knowledge that any errors that arose would be from the new code and not the pre-existing code.

Updating to PRI+

The changes that were made to turn **PRI** into **PRI+** are detailed in Section 2.1.2. For the purposes of versatility, it was decided to implement **PRI+** alongside the existing **PRI** functionality. A Boolean value is used to switch between the two different calculations. Implementing **PRI+** was a matter of rewriting most of the functions for **PRI** to include the regularisation parameter λ and utilise it in the new calculations. A small logic change was also required to implement the new formula for calculating word frequency, $\varphi_{\lambda,\mathcal{D}}(w|a_k)$, which replaced $\phi_{\mathcal{D}}(w|a_k)$, in the **PRI** calculation.

It was in developing this functionality that the discovery of the trivial solution for λ was made, the equations for which are presented in [4]. Alongside this, it should also be noted that the **PRI+** value used in this project is not the normalised version that is finally arrived at in Section 3.3 of [4]. It was unclear how the empirical mean and variance were calculated and thus this project instead uses the **PRI+** score detailed in Section 3.2 of [4]. The values of any results obtained in this project will thus differ from their equivalents in [4], but the overall trends should still be the same, and if the results from the previous research were still to hold, similar evidence should be observed.

A heat map is used to display the perceived user interest according to Google. This heat map is a grid of labels vs. **PRI+** scores as shown in the image below. While normalisation would provide more consistent values from probe to probe, it is possible for users to control the framing of the graph such that only a small subset of probes are visible at a time, which provides a more representative display of user interest in the case where **PRI+** scores vary greatly between probes. The data used to generate the graph is also available to export in JSON format, which is what is used to gather the **PRI+** data for testing discussed in Chapter 4.

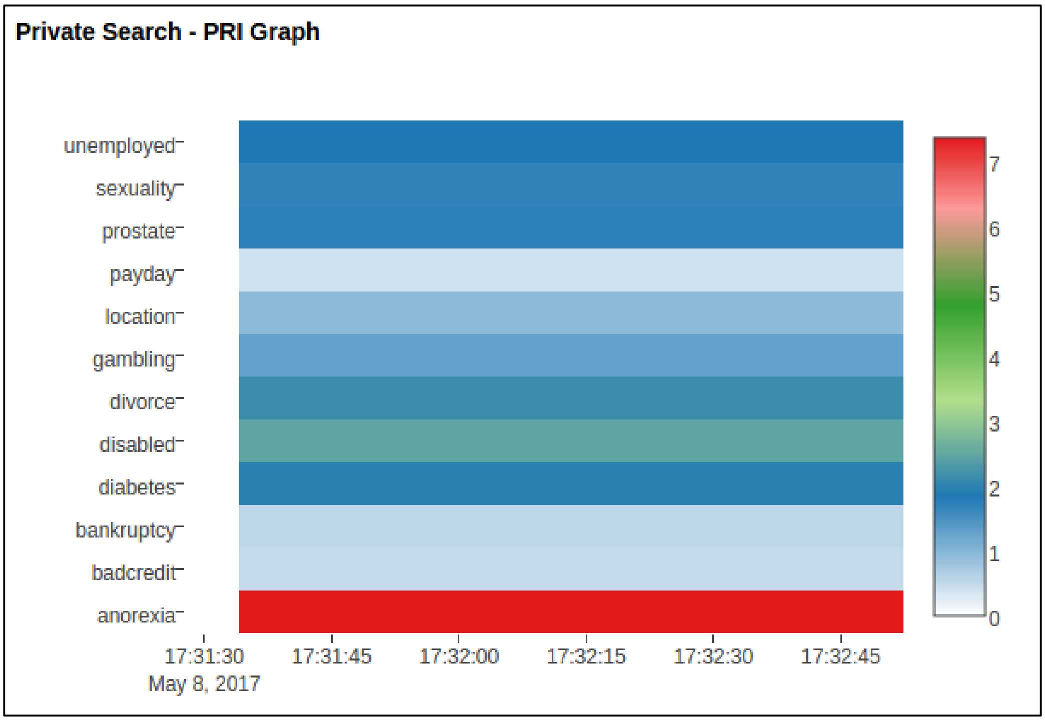


*Image II: **PRI+** Heat Map*

3.4 Training Data

The training data is comprised of a large set of sample adverts. Each of these sample adverts has an associated label which categorises it as one of the sensitive topics.

```
'gambling:: Bettors Club Free Football Tips Predictions with Tipster Statistics;'
```

When the plug-in is initially loaded, the training data is parsed to create a matrix which contains a count of the occurrences of each unique keyword associated with each label. This matrix is then used in the calculation of **PRI+** scores.

To ensure accuracy of classification of ads, both in the user response pages and to evaluate the **PRI+** scores of probe queries, it was necessary to create a new training data set.

The requirements of this new training data set were that it provide a high level of accuracy for each topic and, ideally, to limit misclassification to topics that are similar in the nature of the ads which they return. Finally, it would be desirable for this training data set to have a minimal amount of bias towards any particular topics.

The new training data set was created by running user queries for each sensitive topic and extracting the ads that were shown on the user page in response. In [3, 4], the training data used also included a set of keywords for the ‘non-sensitive’ topic, which were created by taking a set of top searches from popular search engines as examples of uninformative queries and excluding any terms which appear in sensitive topics. This set of keywords was not included in the training data in this project due to the time constraints that were present. While having a set of sample ads for the ‘non-sensitive’ topic would be desirable before releasing the plug-in to the public, it was not a necessity for evaluation of learning within this project, as the majority of ads that were shown in response to user and probe queries could be classified using one or more of the sensitive topics in question. It should be noted that the original training data set did not contain sample ads for the ‘non-sensitive’ topic either, and thus comparison between the two sets will not be affected by the lack thereof in the new training data set.

Using the method described above, a training data set of almost 3,000 sample ads was accumulated. The evaluation of the correct classification and misclassification of this training data set versus the original training data set is discussed in Section 4.3.

The training data set does not contain an even distribution of ads for each topic as some topics provide a much greater volume of ad response than others. While this will lead to a certain degree of bias within the data set, eliminating this bias is not as simple as having an even number of ads for each topic. Rather, the only way for the training data set to have no bias at all is for each unique keyword in the data set to appear an equal number of times.

Unbiased is defined by all topics being equiprobable in the absence of any keywords from the dictionary appearing in the ads shown in response to a probe query. That is, in the absence of any keywords that have previously been encountered, the values of all of the **PRI+** scores should be equal.

Creating a set of training data with this property is infeasible, as it would require a large amount of time and to falsely boost the frequency of certain words. It is also impossible to maintain as users add new data to the set.

In [4], the regularisation parameter λ was proposed to account for this bias. Unfortunately, as discussed in Section 2.1.2, the equations provided in [4] have a trivial solution of $\lambda \approx 1$, and so a constant value of 0.01 is used instead. No solution to this issue was reached during this project, although it should be noted that this bias will only cause issue in two cases. These cases are when no ads occur and when ads that are present contain keywords that are present in a similar frequency in the training data set for two or more topics. No occurrences were observed where this inherent bias caused a misclassification in the results.

3.5 Proxy Topic Injection

As discussed in Section 2.1.3, proxy topics injection was carried out in [4] by carrying out 3-4 proxy queries and 1 sensitive query between each pair of probe queries. This was simple to do in [4] as all testing was scripted. A test session could be created in advance of runtime, with the required number of probe queries, proxy queries and sensitive queries, in any desired order. Unfortunately, this becomes a bit more complicated when there is no control over when a user will carry out a search.

There were a number of different options available in deciding how best to approach proxy topic injection. In order to interleave user, proxy and probe queries in a desired sequence, synchronicity is required. To achieve this it is necessary to carry out proxy topic searches in the foreground of the plug-in, that is, in the user thread. Doing so has certain consequences: the operation of proxy queries is exposed to the user, and the injection of these proxy queries interferes with the user's own session. This was not considered to be an acceptable trade-off, and so asynchronous operation was explored.

As mentioned briefly in the technical challenges in Section 1.3, synchronicity in the background thread of WebExtensions has recently been depreciated due to the potential impact it can have on the user experience. This meant that in designing the proxy query injection, it was uncertain what order the queries would come in relative to the user searches.

Two options remain, both of which were tested. The first is to run the proxy topic injection from the main background thread of the plug-in. While this is functionally possible, it runs the risk of adversely affecting the overall performance of the plug-in. It was found that implementing the proxy queries through the background thread created a bottleneck where the user might search for another topic, and expect the ads on the new response page to be labelled, but the background thread was still carrying out queued proxy queries.

The end solution arrived at is to use a worker thread to operate the proxy queries. Worker threads are JavaScript's equivalent of multi-threading, and allow for proxy queries to run without interrupting either the user session or the background thread’s processing. As this thread functions asynchronously, the logic present in this project is slightly different to that in [4].

It is proposed here that it is not necessary to have a sensitive query between all probe queries which are carried out, and that for the search engine to consider sensitive queries as noise, sensitive queries only need to be adequately interspersed among the sequences of proxy queries. Thus, the worker thread will continually carry out proxy searches, with random intervals in between them, up to a limit of 8 queries without a user search having been carried out. The background thread keeps track of the number of user and proxy queries that are carried out, and runs a probe query after a total of five searches of either type has been carried out.

The random intervals are necessary because Google has measures in place to detect robot probes, and so random delays are required to avoid being hit with a captcha screen. The limit on the number of proxy queries is introduced for two reasons: first, there is a limit on the number of searches which can be carried out in a particular time frame without Google perceiving it to be a DDoS attack; and secondly, it is desirable to make sessions, including proxy queries, appear as natural as possible, and thus it is not desirable to continually carry out searches when the user is not actively interacting with the website. The exact value of the limit was arbitrarily selected, though the impact different values might have could be investigated in future work.

Unfortunately, the effectiveness of the proxy topic injection in this project was not possible to evaluate. This was because results obtained from the detection method did not provide enough evidence of learning for a disruption method to be necessary or useful. These results are discussed in further detail in Chapter 4.

Chapter 4

Evaluation

This chapter details the main tests that were conducted over the course of this project. Each section begins with discussion on the set-up of the test, followed by the results observed and a discussion of these results. Possible sources of error and improvements that could be made are also proposed.

Extensive testing was carried out in an effort to thoroughly evaluate the functionality of the plug-in, as well as to compare with the results of the previous research. This testing took a numerous different forms: manual plug-in testing, whereby the plug-in was installed in a web browser and queries were manually input; scripted plug-in testing, where the plug-in was loaded using Selenium and queries were driven by a Python script; and scripted testing without a plug-in, where the scripts which were used in [4] were run to evaluate whether the results from that paper still held.

In excess of 120,000 user queries, and 24,000 probe queries were carried over a six month period as part of this research. Three different Google accounts were created for the purpose of testing in this project. These accounts were compared against each other and against an ‘anonymous’, i.e. not logged in, user. Sessions consist of a certain number of user queries and probe queries, typically 25 and 6 respectively. Between sessions the browser settings, including history and cookies are cleared. This is based on the assumption that profiling is primarily carried out through the use of cookies, and thus any profiling which occurs should only be based on the current session. Tests were run on two different operating systems, Ubuntu 16.10 and Windows 10. Two different networks were used to carry out tests, a personal home network and a university network. Neither differences in operating systems or networks produced a noticeable change in results.

4.1 Ad Response Volume

Setup

One of the issues encountered near the beginning of this project was that ad response seemed to be very low in comparison to that reported by the authors of [4]. In particular: as the length of a session increased, the volume of ads shown in response appeared to decrease; and any ad response for the probe queries used in [3, 4] seemed sporadic at best.

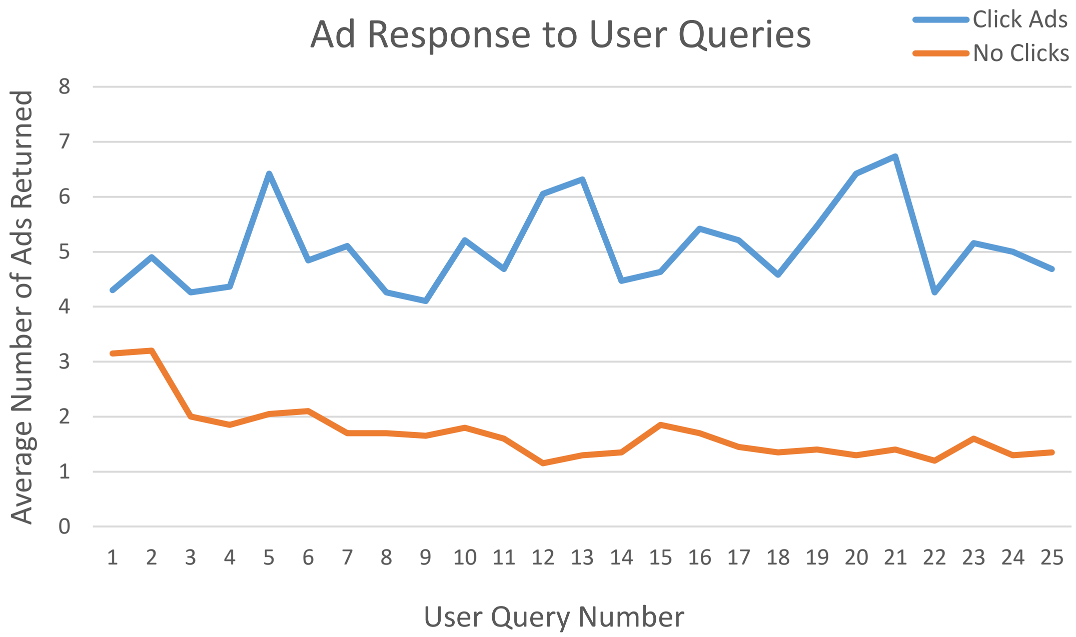
Assumption 2 discussed in Section 2.1.1 relies on ads to be shown in response to user queries and probe queries to be of use. Another assumption is proposed here for consideration. That is, a for-profit commercial recommender system, such as a search engine, will provide a greater volume of ads to a user who is considered more likely to interact with those ads and thus more likely to generate click through and sales. The likelihood of interaction is measured by the user's prior interaction with links or ads which have been provided.

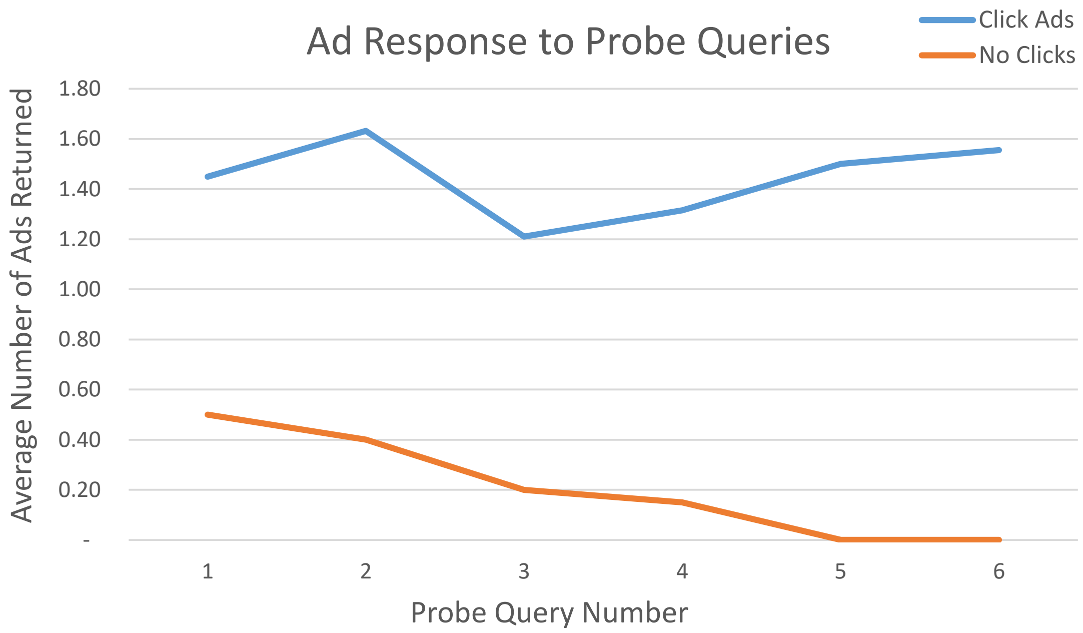
To investigate whether this assumption was valid, tests were carried out to compare the volume of ads shown in response to both user and probe queries. Three tests cases were examined: the case where the user clicks neither ads nor links; clicks only on relevant ads; or clicks only on relevant links. It was decided to only test these three cases as they were deemed least likely to obscure user interest in the search topic. To fully analyse the ad response for user interaction, it would be necessary to use a number of different click strategies, such as those used in [4].

The same measurement of 'relevance' is used in these tests as was used in [4] for ads and links. The user queries were evenly distributed across all sensitive topics. These tests were run for an 'anonymous' user and three different logged-in users. In total, 7,200 user queries and 1,440 probe queries were carried out for each of the click strategies.

Results

The following graphs show the variation of ad response over the course of a user session. In this set of tests, user sessions consisted of 25 user queries and 6 probe queries, with the probes placed before and after every 5 user queries.





From the above graphs it can be seen that when a *no click* strategy is used, ad response for both user queries and probe queries decreases over time. While the ad response to user queries is relatively unimportant for the techniques used within this project, the lack of ad response to probe queries means that it is not possible to detect any learning using the methods discussed in this paper. As discussed in Section 2.1.3, [4] found that different click strategies had little effect on the user-profiling which was occurring. Here it is shown that a click strategy might now be necessary for the measurement of that learning to be possible in the first place.

In contrast to the *no click* strategy, *click relevant ads* provides a consistent ad response across both the user queries and probe queries. These results could suggest one of a few possibilities. With the *no click* strategy, there is no interaction with the results page other than the search queries that are input. While it is possible that the results page might provide a user with the information which they were looking for, it is probably atypical for a user to have no interaction over many different searches. Hence it could be that, even with random intervals between searches, Google deems the session to be automated and stops showing ads.

Alternatively, Google might categorise users based on their likelihood of clicking an ad, as proposed in the assumption discussed above. Accordingly they show more ads to users who are deemed more likely to click them, leading to the results shown for the *click relevant ads* strategy. Unfortunately, what is deduced from these results will come down to the assumptions made about the workings of the search engine, with no guarantee that those assumptions are actually correct.

It should be noted that although three test cases were specified, *no click*, *click relevant ads* and *click relevant links*, only *no click* and *click relevant ads* are present in these results. Unfortunately there was difficulty encountered in collecting data for the *click relevant links* case. Examining the log files showed that the automated scripts were crashing due to time out errors. Discerning the exact cause of these crashes was difficult within the time frame of this project, although there are known issues with Selenium and the Chrome WebDriver which could be responsible, [30] in particular.

The purpose of this set of tests was to determine a method by which ad response could be improved, and consequently allow for the analysis of learning. As such, it was decided that the results obtained by the two cases shown were sufficient and manual testing for clicking of links was not conducted. From the partial data that was obtained through automated testing, *click relevant links* appeared to provoke a greater ad response than the *no click* strategy, but a lower ad response than the *click relevant ads* case.

The fact that information is lacking regarding the inner workings of the search engine means that there could be unforeseen effects of using any of these click strategies. Ideally, a set of tests should be carried out using multiple different click strategies, measuring both the ad response and the variation in the ads which are shown. This would allow for determination of whether the click strategies influence the personalisation of the search engine.

4.2 Probe Query Selection

Setup

In [3, 4] two probe queries were used; ‘help and advice’, which was deemed to be reasonable for use as a probe query for any sensitive topic, and ‘symptoms and causes’ which was deemed to be reasonable for use as a probe query for medical topics. As mentioned above, the ad response to these probe queries in this project was very low by comparison with those reported in [3, 4]. While [4] indicated that after 3 or 4 probe queries in a 5 probe session learning was evident, in this project it was found that at most one probe query in a 5 probe session might return any ads at all, and these ads were not guaranteed to indicate any learning.

To investigate whether the lack of ads was due to using a probe query that was not informative or specific enough for the search engine, a number of different probe queries were trialled within this project. The probe queries selected for testing were ‘medical advice’, and ‘find free help’. ‘Medical advice’ was selected based on the expectation that including the word ‘medical’ might provoke a greater ad response than ‘symptoms and causes’. ‘Find free help’ was selected by compiling a list of terms which appear at least once for all categories in the training data and then selecting three which occur in high frequency and make sense together linguistically. *Table I* below shows the relevant statistics which were used in the selection of these words.

These probe queries were tested against the original two probe queries, and evaluated based on the volume of ads returned and on their performance as probe queries. It was not feasible within the scope of this project to run comprehensive tests across all topics for each of these probes queries. Instead any bias which is present in each probe query is evaluated through the use of **PRI+** scores. Significant bias towards a particular topic which is unrelated to the current topic signifies a poor probe query. The user queries in these tests were two queries related to ‘prostate’ that each generated a high ad response. These queries were alternated with each user search.

Word	Average	Min	Max	Total
now	61.17	3	166	734
find	54.17	2	180	650
free	46.67	2	163	560
about	44.17	2	173	530
you	39.42	6	116	473
with	38.17	1	110	458
types	32.83	1	124	394
help	31.00	1	150	372
dublin	28.42	1	126	341
more	23.67	1	77	284
new	14.41	1	62	173
services	13.58	1	45	163
visit	11.17	1	32	134

Table I: Analysis of words which appear in every category of the Training Data

Results

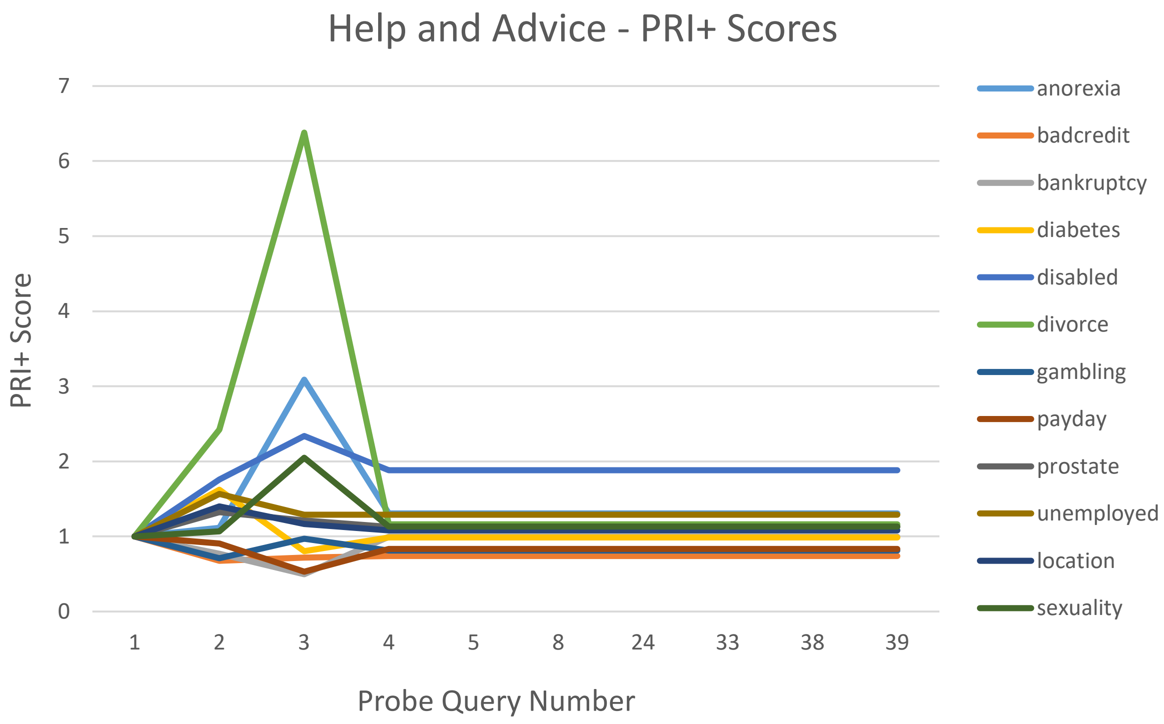
The following table shows the percentage of probe queries that had an ad response, for *no click* and *click relevant ads* strategies.

Probe Query	No Click	Click Ads
help and advice	0.00%	23.08%
medical advice	2.57%	100%
find free help	0.00%	56.41%
symptoms and causes	2.57%	100%

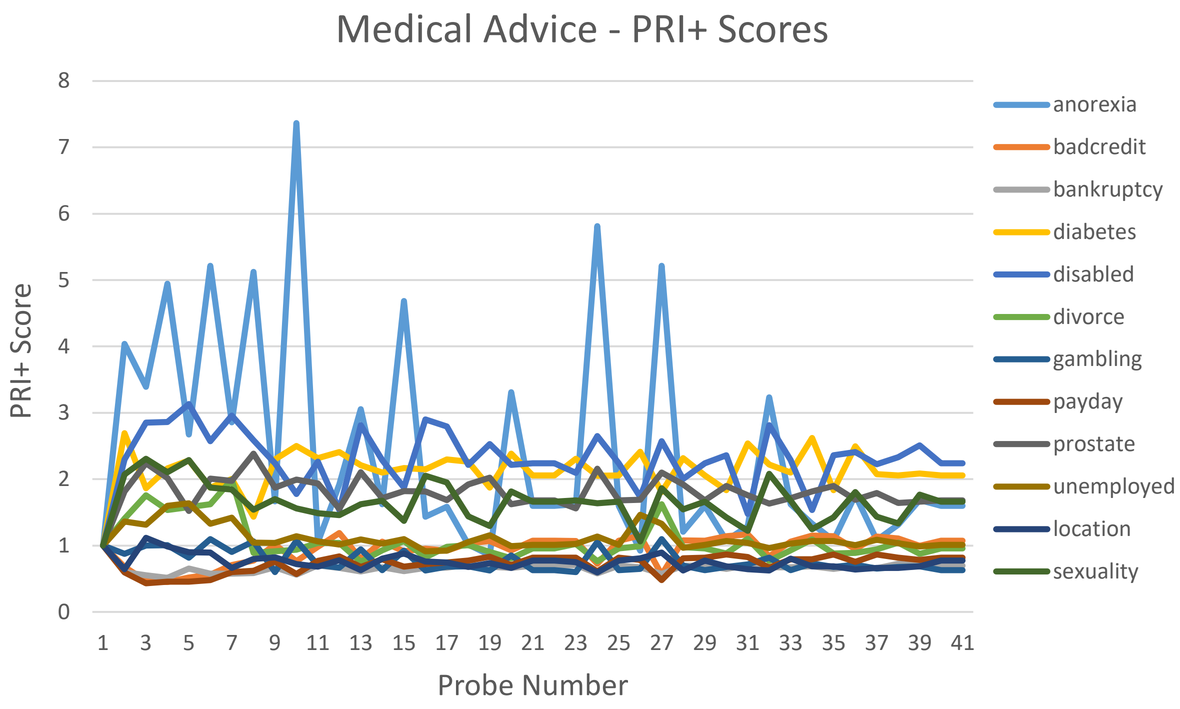
Table II: Percentage of probes with ad response

From the above data, ‘symptoms and causes’ and ‘medical advice’ provide the best ad response, in both the *no click* and *click relevant ads* cases. The very low ad response in the *no click* case is consistent with the results from Section 4.1. There is an inherent bias within these results as the topic which was searched for is a medically related topic. To obtain a more fair set of results, this test should be repeated across all topics. It is expected that the same trend will be present for all medical topics, and that for non-medical topics, the ad response for 'medical advice' and 'symptoms and causes' is likely to drop. It is difficult to predict how the ad response to other two queries might change for non-medical topics, although the results in [3] suggest that the ad response may improve for 'help and advice'.

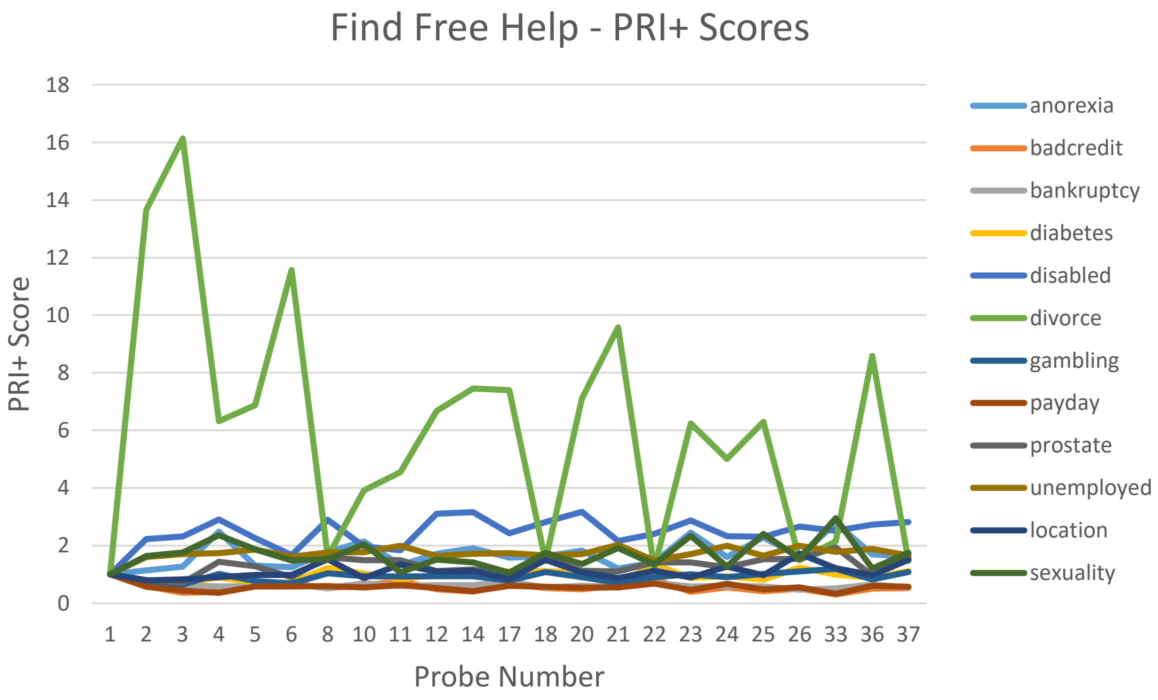
Next, the bias present in these probe queries is analysed. A bias is considered significant when there is a spike in the **PRI+** score for a topic unrelated to the topic being searched for. It is expected that any such bias would be present regardless of the presence of learning, i.e. the probe query itself is likely to turn up ads related to the biased topic whether learning is occurring or not. In the graphs below, only data points for the probe queries that have ad response are plotted, as these are the only points in which information about the probe query is present.



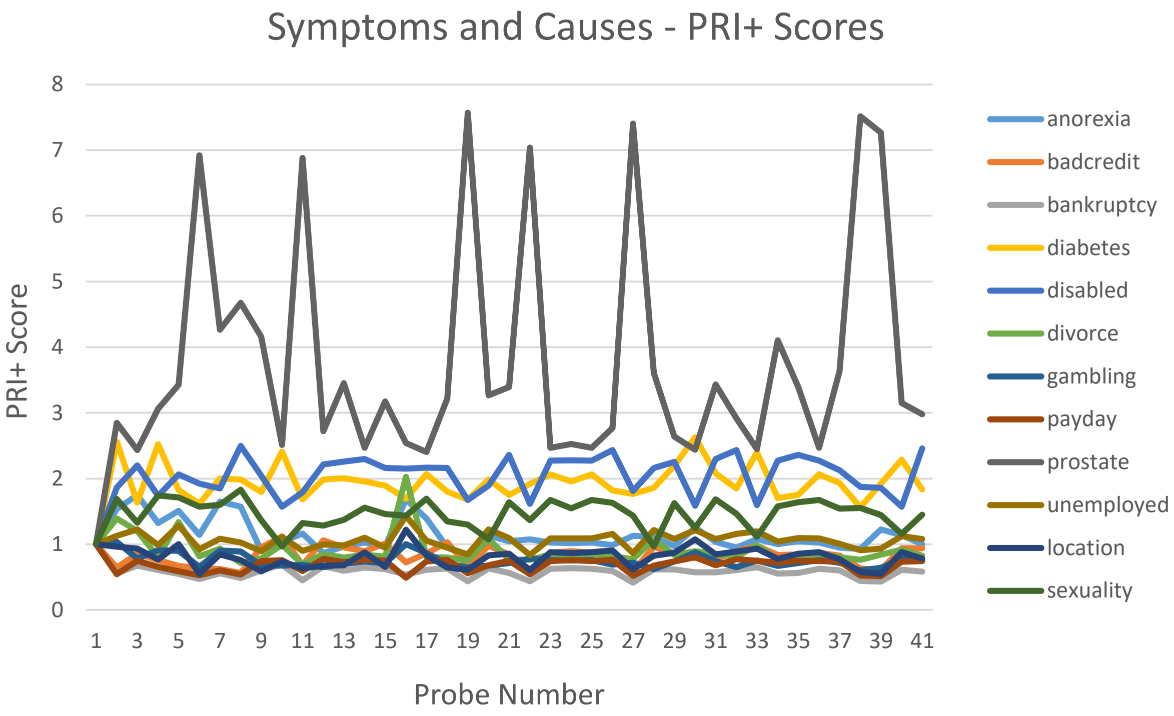
From this graph it can be seen that 'help and advice' causes an initial spike for 'divorce' and 'anorexia' but following that it levels off. The ads which are initially shown in response to the 'help and advice' query are legal ads, and thus 'divorce', which has a large selection of legal ads in the training data, comes up with a bias. This bias is relatively low compared to some of the other probe queries, although there is no indication of any learning towards prostate.



For 'medical advice', a bias is present towards all medical topics. The most prominent categories specifically are 'anorexia', 'disabled' and 'diabetes'. The fact that 'medical advice' has a bias towards medical topics means that it is potentially useful for those topics, but not necessarily useful for any non-medical topics. On top of this, the bias towards certain medical topics needs to be accounted for when trying to determine evidence of learning.



The above graph shows an overwhelming bias towards 'divorce' for the 'find free help' query. This query was selected by looking at common words in the ads in the training data, in the hopes that it might provoke a greater ad response, but it appears that the choice of words leads to an excess of legal ads. These results suggest that 'find free help' is a very poor probe query to use in the context of this project.



The 'symptoms and causes' probe is the only probe which provides any evidence of having determined the topic which was being searched for in the user queries. Given these results, and the results of the ad response evaluation, this probe query seems to be the most ideal. Unfortunately, these results have only been evaluated for a single topic, and thus do not give a good indication of the utility of this probe for other topics. It is expected that for other medical related topics, the same trend would appear.

Given the limitations of the testing carried out, it is hard to make a judgement on the best probe query to use. These results suggest that 'symptoms and causes' might be the best for medical queries, but further testing is needed to confirm this. The one thing that can be said with certainty is that 'find free help' appears to be a particularly poor query for the categories that are being examined in this project.

In light of the evidence of learning apparent in this final graph, a new set of tests were carried out to analyse whether there was evidence of learning for a typical user session. It should be noted that the two user queries used in this test, '*prostate cancer men over 50'* and *'treatment for early prostate cancer'* were selected because they each provide a high ad response. In a typical user session, where the number of ads will vary greatly, and the average ad response is lower than in these tests, it is expected that the **PRI+** score for ‘prostate’, i.e. the evidence of learning, would be less.

4.3 Training Data

Setup

In order to compare the two sets of training data, manual testing was carried out as follows. A query was input, and the ads on the resulting page were categorised manually, and compared with the suggested category returned by the plug-in. Given that there is no guarantee that ads returned in response to a query will match the category of that query, it was decided to simply categorise results as either true or false, where true means that the manual categorisation matched the suggested category returned by the plug-in, and false means that the manual categorisation did not match the suggested category returned by the plug-in.

Between each query, the web browser was closed and re-opened, and the history and cookies were cleared. This ensures that the previous searches have a minimal effect on the results of subsequent searches, which is particularly important when subsequent searches are across different topics. For each topic, 30 queries were run, using the same lists of sample queries that were used in the scripted testing of the plug-in. In the case where a list had less than 30 queries, the list repeated from the start one the end of the list was reached.

One shortcoming of this method arises in the case where manual categorisation of the ads is ambiguous. This is only the case when multiple categories return similar ads, for example in the case of categories; bankruptcy, payday and bad credit. In these cases, unless the ad specifically mentioned words associated with a different category, it was assumed that the category was relevant to the current topic being searched for. This shortcoming could have partially effected the results obtained, although using the same method for both sets of training data should mitigate the effect to a certain degree.

Results

The following table provides a comparison between the original training data, which was provided as part of the pre-existing plug-in, and the training data that was created as part of this project. The figures shown are the percentages of ads that were correctly labelled by the plug-in using the specified training data sets.

Topic	Old Training Data	New Training Data
anorexia	9.52%	100.00%
bad credit	11.43%	35.24%
bankruptcy	44.25%	38.94%
diabetes	10.00%	100.00%
disabled	0.00%	87.50%
divorce	89.66%	89.66%
gambling	64.36%	77.45%
location	20.79%	98.02%
payday	0.00%	61.39%
prostate	46.00%	96.00%
sexuality	12.68%	87.32%
unemployed	17.31%	84.62%

Table III: Percentage of correctly labelled ads in response to user search queries

A number of observations were made when gathering the data for the above table. Firstly, within individual topics, the queries that were run returned a similar number of ads. Similarly, running the same query more than once typically returned the same number of ads. From this it can be concluded that the resulting values are fairly consistent and that the differences in results are primarily due to the differences in training data.

Next, the number of ads that were displayed for a particular topic, based on 30 queries, varied greatly between topics. While a detailed analysis of the reasons why is outside of the scope of this work, it is not unreasonable to assume that topics with more revenue potential will generate more ads. Similarly terms such as ‘disabled’ and ‘anorexia’ might return fewer ads due to the perceived negative connotations associated with them. The three topics with the lowest ad responses were ‘anorexia’, ‘disabled’ and ‘diabetes’, all of which had between 20 and 30 ads displayed in total.

Ideally to give a more statistically significant result further queries should be run. For the purposes of comparing the two training data sets, even this small sample size makes the trends clear. The number of ads shown in response to each training data set for particular categories was not always identical, and so to provide as equal a comparison as possible, the number of ads taken into account was the lower of the two amounts.

It was decided to provide a conservative estimate of the performance of the *new training data* relative to the old. In the case where the *new training data* had a greater ad response, the difference was taken from the number of correctly labelled ads. In the case where the *old training data* had a greater ad response, the difference was taken from the number of incorrectly labelled ads. While this does mean that the results for both sets of training data are not entirely accurate, the intent was to provide a worst-case scenario comparison between the two sets.

It was also of interest that there was very little run-over between topics. After running 30 queries for one topic, the first query of the next topic did not return any ads related to the previous topic, even in the cases where topics had a similar overall theme, such as health. While this could be attributed to the clearing of the browser settings discussed previously. It could also be an indication that Google's algorithms assess user behaviour on a larger scale, by comparing it statistically with a very large data set, rather than analysing individual trends.

The only case in which the *old training data* provides better classification than the *new training data* is in the case of 'bankruptcy'. In this case it is important to investigate how ads were misclassified. For the *old training data*, 36.28% of 'bankruptcy' ads were misclassified as 'non-sensitive', 13.17% as 'bad credit', and the remaining 6.19% as 'anorexia' or 'divorce'. For the *new training data*, 40.71% of 'bankruptcy' ads were misclassified as 'payday', 14.16% as 'bad credit' and the remaining 6.19% across four other topics.

It is clear from the above results that the *new training data* provides better classification of ads overall. With the *new training data* there are only three topics that provide categorisation accuracy of less than 77%. The three topics in question, 'bad credit', 'bankruptcy' and 'payday', are closely related to each other. In the cases where these topics were mislabelled, they were most commonly mislabelled as one of the other two similar categories. It is safe to assume that if categories were made more distinct from one another, or these categories were merged into a single category, that the classification would follow a similar trend to all other categories in the list for the *new training data*.

Furthermore, it is not clear from [3, 4] exactly how the topic queries for the scripted sessions were selected. To increase the reliability of the results and allow for more accurate classification of similar topics, it would be useful to provide a group of participants with a set of topics to research over a number of days and gather the queries that they search for, as well as the ads that they are presented with in response. This would not only enhance the training data further, but it would also provide a list of queries for use in the scripted tests that accurately reflect real world behaviour.

4.4 Testing for User-Profiling

Setup

As stated in Section 2.1.2, it is assumed that it is possible to detect evidence of search engine profiling by examining the advertisement content in response to probe queries injected into a stream of user queries. This profiling is measured by using the **PRI+** score detailed in Section 2.1.2. There is considered to be evidence of profiling when the **PRI+** score is noticeably higher for the topic which was being searched for in the user queries.

Testing for user-profiling has occurred throughout the course of this project. Within all of the other tests carried out, the history of **PRI+** scores are exported and analysed. While this data is valuable in that it can provide confirmation or rejection of the expectations of the project, summarising this data is difficult as all of the tests were carried out under different conditions.

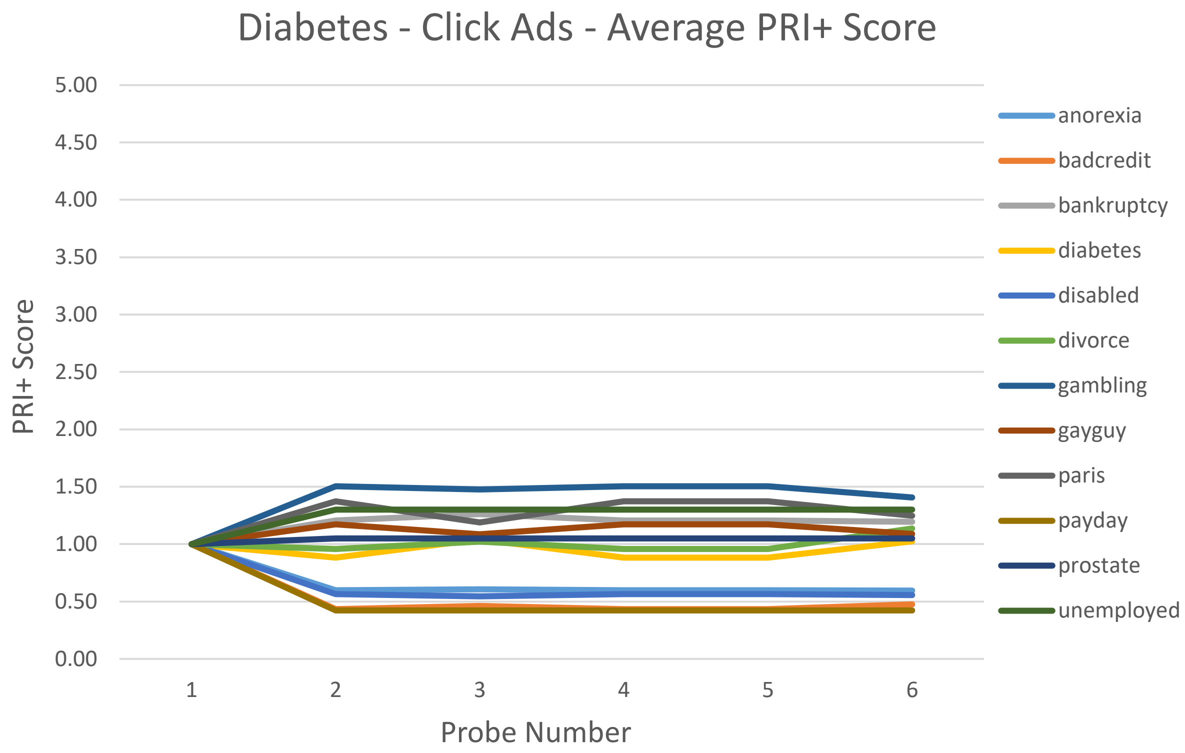
In order to explicitly test for user-profiling, a number of different scenarios are examined. As mentioned in Section 4.1, ad response for user and probe queries was unexpectedly lower than that reported in [3, 4]. A click strategy is introduced to improve this ad response, but it is necessary to consider the impact the clicking has on the profiling, or lack thereof, carried out by the search engine.

Originally a set of tests using two different click strategies, *no click* and *click relevant ads* were ran in March 2017. These tests consisted of a total of 600 user queries and 144 probe queries for each topic, split evenly between both click strategies. These queries are further split into sessions which are 6 probes long. As such 12 sessions worth of data were obtained per topic for each click strategy.

In light of the results obtained in Section 4.2, which were carried out at a later date than the original tests for user-profiling, another set of tests was carried out in the early weeks of May 2017. Due to the short time frame remaining in the project, the breadth of these tests was limited and thus tests were only carried out for two topics, ‘prostate’ and ‘diabetes’.

Results

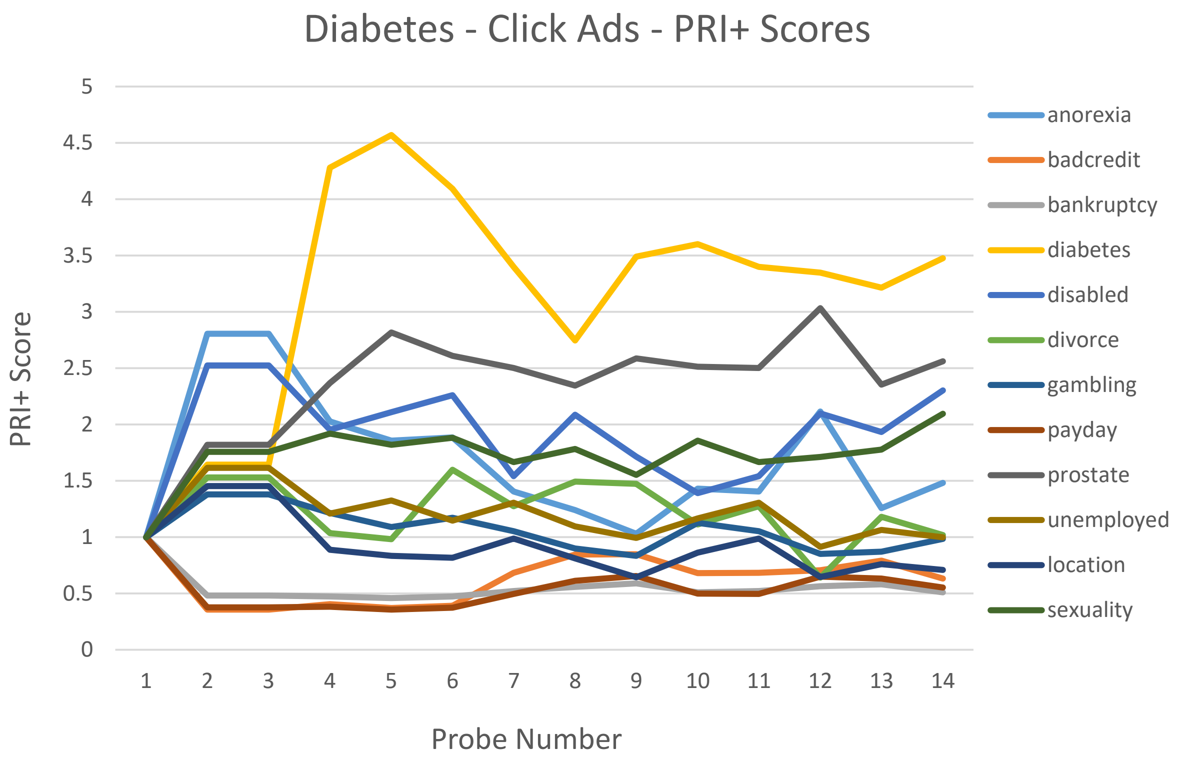
In the first set of tests that were carried out to test for user-profiling, no evidence of learning was observed. The set of tests in which there were no clicks returned very minimal ad results, in keeping with the results in Section 4.1. For the tests in which ads were clicked, results were all consistent with the graph shown below. It should be noted that the y-axis in this graph has been extended up to a value of 5 to provide a basis for comparison with results further on in this section.

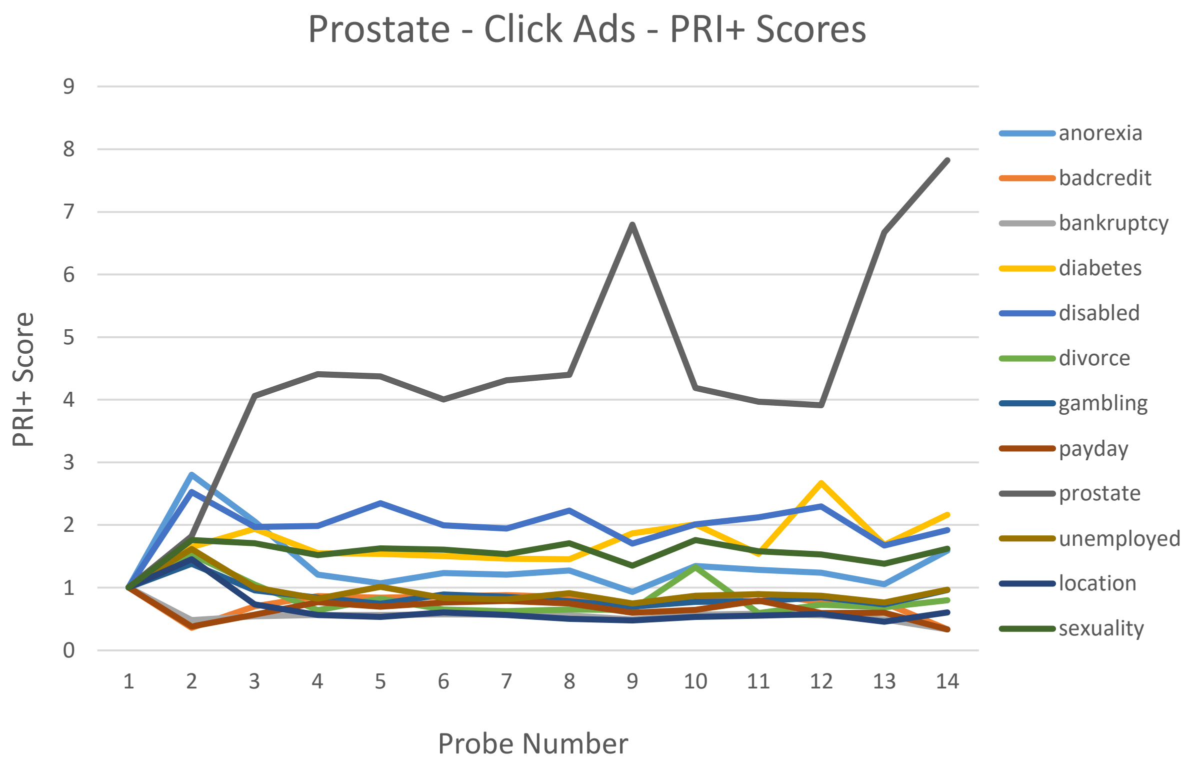


From the graph it can be seen that there is no evidence of learning of the sensitive topic, 'diabetes', and that overall there is relatively little variation in the **PRI+** scores. The four topics that are centred slightly below the others can be attributed to the fact that those topics had fewer keywords associated with them, and so there is an inherent bias present in the training data against those particular categories.

The values present in this graph are the result of averaging the **PRI+** scores across the 12 sessions that were carried out for 'diabetes'. The trends in this graph were present across all topics that were tested. The same set of tests were carried out for a number of different probe queries, namely 'help and advice', 'symptoms and causes' and 'medical advice', with similar results in each case. These results, in conjunction with these same trends reappearing across all other tests, seemed to indicate that no profiling was occurring.

After testing the effectiveness of different probe queries, as discussed in Section 4.2, it was decided to re-test for evidence of user-profiling. The following graphs are the results of this re-test.





It is apparent that after 2-3 queries in each of these cases, there is clear evidence of learning. In both cases, there is an overwhelming increase in **PRI+** score for the topic which is being searched for in the user queries. To confirm that these results were not just a single occurrence, the tests were repeated for each of the three user accounts as well as anonymously. All tests bar one, the anonymous run searching for 'diabetes', followed the same trends displayed in the graphs above. Repeating the tests without clicking on ads resulted in poor ad response, and thus no change from the prior results is observed.

Unfortunately, the only conclusion that can be drawn from these results is that ads are being shown in response to a "relatively neutral" probe query and that these ads correspond to the topic which was being searched for in the preceding user queries. Relatively neutral here means that it does not indicate interest in any specific topic, although it may indicate interest in a subsection of the overall topics.

There is still no evidence of learning without clicking on the ads that are present in the response page. This could be an indication that it is the user interaction through clicks, rather than the searches that are being conducted, which Google is using to learn. This would contradict the results found in [4] with regards to the effect of click strategies.

Another alternative, and possibly an explanation for the variation between the two sets of results obtained within this project, is that Google largely ignores both the user queries and user clicks on *google.com.* Rather, they could primarily carry out their tracking and profiling through users visiting external websites. Unfortunately, the ads present on the user response page were not recorded for the first set of tests that were carried out, and thus they cannot be compared with those present in the later sets. It is possible that these two sets differ and that tracking was occurring through the latter set. This would be consistent with the findings of [6].

Without gaining access to the source code for the algorithms that decide which ads are displayed in response to a particular query, it is difficult to determine the exact reason for these results. A more comprehensive test set is thus a necessity in this regard.

It is unclear exactly what contributed to the difference in results obtained between March and May. The only differences between the tests carried out were the inclusion of the new training data set. The ads were manually examined in the case of both sets of tests, and there were clear differences in the types of ads being shown in both cases. The differences could be due to a change made by Google which is not visible to users, but without further testing this is just speculation.

4.5 Miscellaneous Tests

In this section, a brief description of some of the smaller tests which were run is provided. These tests were carried out in order to confirm certain assumptions that were made throughout the project. In particular these tests relate to the assumptions regarding the methods by which a search engine such as Google carries out user-profiling.

4.5.1 Manual vs. Scripted Testing

Setup

At the start of this project, one question which was present was whether a search engine such as Google could determine the difference between queries run through a scripting application such as Selenium, and queries run manually through a web browser such as Chrome. It was decided to use the Chrome WebDriver for Selenium rather than a headless WebDriver to minimise any differences that might be discernible between the two.

In order to confirm that these two environments have negligible effects on the user-profiling which is being done by the search engine, a session was run consisting of 25 user queries and 6 probe queries for each topic as an ‘anonymous’ user in a scripted environment and in a manual environment. The resulting **PRI+** scores for each topic are then graphed and a comparison is drawn between the two. A certain amount of fluctuation is be expected as it cannot be guaranteed that profiling will always occur at the same rate, nor that the ads shown in response to profiling will always be consistent. Differences are considered to be significant if a particular topic, or set of topic, stands out in one set of data and not in the other.

Results

There was no discernible difference present between manual and scripted testing of the plug-in. When no click strategy was used, there were corresponding decreases in ad response for both user and probe queries in both scenarios. In the case where relevant ads were clicked, there was an increased in ad response for user and probe queries.

The **PRI+** scores in both cases were consistent with the first set of results in Section 4.4. That is, across all topics, there were no significant differences in **PRI+** scores between the two data sets. Any differences that were present could be attributed to slight differences in the ads which were present in response to the probe queries. These differences are in keeping with the low consistency of ads discussed in the next section. The only significant trends within the **PRI+** scores were those that occurred due to the bias in the training data, and not due to any notable presence of learning.

Overall, these results appear relatively positive with regards to the requirements of this project. That is, they confirm that it is possible to carry out automated testing of the plug-in, and safely assume that those results will hold for the case of manual use of the plug-in by a consumer. These tests were carried out prior to the appearance of learning discussed in Section 4.4, and thus require re-testing to confirm that they still hold.

4.5.2 Consistency of Ads

Setup

The key assumption in this project is that is it possible to detect and assess search engine learning by analysing the differences in advert content which is shown in response to probe queries. To confirm that these differences are primarily in response to user-profiling by the search engine and due to some element of randomness, the consistency of the ads shown in response to a number of different queries is analysed.

Originally the probe query ‘symptoms and causes’ was selected, along with one query related to each of ‘prostate’ and ‘diabetes’. 100 searches were carried out using each of these queries, and the ads that were shown in response to them recorded. Unfortunately, ad response for these queries proved to be insufficient in determining the consistency of ads.

To compensate for this, an attempt was made to increase ad response by introducing a click strategy, namely *click relevant ads*. Another set of tests of 50 searches of each query was run this click-strategy in place. Finally, a test was run using two different queries related to ‘prostate’, alternating between each of them, with clicking of relevant ads still in place. The ads shown in response to each of these queries were separately analysed.

Results

Following a similar trend to the results in Section 4.1, the original tests to investigate ad consistency were hampered by the fact that ad response was incredibly low. For the ‘symptoms and causes’ query, ad response was 6%, with two different ads, each appearing in 3% of the queries. Both the ‘prostate’ and ‘diabetes’ queries had an ad response of 2%. Ad consistency in this 2% response was 100%, but 2 queries is not significant enough to draw any conclusions from these results.

The introduction of a click strategy provided little improvement of these results. In all cases, there was no immediate ad response, and thus the consistency expected from the click strategy was not possible to achieve. From all of these results it can be assumed that repetitively searching for the same query without making any other searches appears unusual to the search engine and thus the search engine does not provide a significant ad response.

The final test carried out, alternating between two different prostate queries, provided a significant improvement in ad response, supporting the above assumption. The two queries used were '*prostate cancer men over 50*' and '*treatment for early prostate cancer'*. Ad response for these queries was 92% and 96% respectively. The table below shows a summarisation of the data obtained from this test.

Query	**Total Ads**	**Unique Ads**	**Average occurrence of each ad**	**Unique Domains**	**Average occurrence of each domain**
prostate cancer men over 50	89	40	2.22	20	4.45
treatment for early prostate cancer	104	54	1.93	33	3.15
Total	193	82	2.35	39	4.95

Table IV: Ad analysis for two queries related to 'prostate'

Of particular interest in this table is the number of occurrences of each ad. Given a session of 25 searches for each query, a particular ad occurs in just under 10% of response pages on average. This is noteworthy because it means that even for quite specific queries, there is a huge amount of variance in the ads which are shown. The fact that there is such variation in the ads shown in response to queries for a single topic needs to be taken into consideration when attempting to use variation as evidence of learning.

While there is clear evidence of learning in [3, 4], the fact that there appears to be a random element to the decision about which ads are displayed is not accounted for in the present version of **PRI+**. This is a difficult thing to account for without access to the source code driving the search engine and its decision making processes.

It needs to be remarked that the data here is based on a relatively small sample set, and that to get a more accurate picture of the variation of ads in general, a larger selection of queries, across all topics should be tested.

4.5.3 Profiling Across a Network

Setup

This test was run to confirm that a search engine such as Google profiles a user through the use of cookies stored in a particular browser, rather than analysing the user queries coming from a particular IP address. To test this, two different machines were used, both running on the same network, and using the Ubuntu 16.10 operating system. On one machine, 100 queries searching for topics related to ‘prostate’ were carried out. At the same time, on the other machine, 100 queries searching for ‘symptoms and causes’ were carried out.

A comparison is made between the ads which are shown in response to the queries for ‘symptoms and causes’ and those obtained during the *Consistency of Ads* tests. Any notable differences in ad response, particularly ad response related to ‘prostate’ would be an indication of profiling occurring across multiple users on the same network.

Results

There was no evidence of any profiling being carried out across multiple machines on the same network. It was found that the ad response for ‘symptoms and causes’, which was run exclusively on one of the machines, turned up almost identical results to those detailed in the *Consistency of Ads* section above. That is, a very low ad response is observed, with only two different ads being displayed, which were unrelated to ‘prostate’.

If profiling across the network were to occur, it is expected that these results would have some skew towards prostate, similar to the results obtained for ‘symptoms and causes’ in Section 4.2. These results support the assumptions made concerning the methods by which a search engine carries out user-profiling, i.e. that it is largely based on the user’s interaction with a website during a particular session.

4.6 Discussion

The main conclusion to be drawn from these results is that user-profiling is clearly more nuanced than can be accounted for with only a small set of assumptions. Even in the case where “positive” results were observed it is unclear as to whether the assumptions made in this project are valid, or if these results are due to a factor outside the scope of this analysis.

These results seem to indicate that user-profiling, and consequently personalisation of search results, does occur, but that this personalisation is not exclusively in response to search queries being carried out. Rather, user interaction of some sort is required for the personalisation to become visible.

It is unclear from these results as to whether that user interaction is merely clicking on a link, whether it is the visit to particular webpages, or some unconsidered factors at work. What is made clear by these results is that even in the case where this user interaction occurs, it will not always be apparent without an adequate probe query. Thus, more time should possibly be spent on determining how probe queries can be built which provoke personalisation of advert content.

Overall, these results show that the analysis carried out in [4] and the findings within are now out of date, and that new analysis needs to be done to determine when and to what extent user-profiling is carried out by search engines like Google. The authors of [4] were kind enough to re-run their tests in light of these results, and their consequent findings confirm those outlined here.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The overall objective of this project was to create a plug-in for use in commercial web browsers which allows users to measure and disrupt search engine learning. While this was not achieved in its entirety, the work outlined in this report provides valuable information for future work in this field.

Four measures of success were established in Section 1.2. The first three of these measures were clearly accomplished as described in Sections 3 and 4. The functionality of the plug-in has been further developed to provide robust and reliable operation. The ability of the plug-in to accurately categorise adverts has been greatly improved through the creation of a new set of training data and implementation of **PRI+**. A comprehensive evaluation of the plug-in's ability to detect and assess search engine learning was carried out, based on the assumptions which were made within this project.

The final measure of success, disrupting search engine learning through the use of proxy topic obfuscation, was unable to be achieved. The functionality for this was implemented and tested to the fullest extent possible. As the techniques used for detection and assessment of learning as described in [4] provided no evidence of learning until the weeks leading up to the end of this project, it was not possible to evaluate the effect proxy topic obfuscation had on this learning.

This work has shown that the techniques described in [3, 4] are not robust to changes made by Google. The techniques rely heavily on the accuracy of the assumptions made within those papers. While the results in those papers show strong evidence to support these assumptions, the results obtained here seem to indicate that those assumptions, in particular ***Assumption 2*** as described in Section 2.1.1, no longer hold in general. The results outlined in Section 4.4 suggest that these assumptions may hold in certain circumstances, but further testing is required to confirm this.

The concept of treating a search engine like a black box is necessary when there is very little reliable information available on the inner workings of the system. However, this simplification can be a hugely limiting factor. It is difficult to confirm any assumptions that are made about the system when only the input and output to that system are available. Furthermore, it is very easy for confirmation bias to be present when analysing any results that are obtained particularly if the tests that are conducted only account for a limited subset of factors.

The tests that were carried out over the course of this project were intended to be as comprehensive as possible, but each has their limits. The biggest limits to these tests were time and scope. Many of these tests could potentially be used as research topics in and of themselves, and may provide greater insight into the inner workings of search engines like Google. The difficulty with any such research lies in the fact that the algorithms behind Google search are ever evolving and utilise massive data sets, the likes of which are unobtainable for all but the most well-funded of research. As was seen in this project, changes can occur which are invisible to the user, but completely invalidate any results which have been obtained.

In conclusion, understanding the prior assumptions that are being made and confirming the validity of these assumptions is an important starting point for any research that is being conducted. Unfortunately, the playing-field is rigged in the favour of those who provide the personalised service, as their work is typically closed-source and rarely publicised. In contrast, the work being done to combat personalisation is often open-source and documentation made publicly available. Therefore service providers such as Google can easily react to any counter-measures which are created with in-depth knowledge of what they are working against, a luxury which is not present for the other side.

5.2 Future Work

With regards to the software development within this project, the end result was a plug-in which functions reliably and robustly. The accuracy of the plug-in was greatly improved through the creation of new training data, but there is room for further improvement. The method used to create training data within the project was limited by the quantity and variety of the queries which were used to generate it. As discussed in Section 4.3, the quality of the training data could be further improved by using a group of real users to create the query lists for each topic, and extracting the training advertisements from their sessions. Furthermore, the catchall category, ‘non-sensitive’, was excluded from the training data in this project. Including this in future training data could easily be done using the same method as described here for sensitive topics.

It is shown that the equations for the regularisation parameter λ in [4] have a trivial solution, and thus this parameter does not fully serve its purpose in the calculation of **PRI+** scores within this project. Additionally, normalisation was not included within this project. As the results in Section 4.4 show, the **PRI+** calculations can clearly be used to detect evidence of learning when it is present. Creating a new set of equations for calculating λ and including normalisation would improve the accuracy of this assessment by reducing the inherent bias in the training data and the ads which are shown in response to a particular query respectively.

It was apparent when attempting to evaluate the truth of the assumptions within this project that there much capacity for further testing. The assumptions which are made in this project are limited in scope to *google.com* and associated domains. While this greatly reduces the set of test cases required, it can also very easily lead to oversimplification. In future work, it would be worthwhile to expand the scope of the project to consider the effect of site visits as a distinct feature from user clicks. That is, compiling a list of sites that are visited in a particular session where ads or links are clicked on, and separately visiting those sites without using Google as an intermediary. This would provide further insight into the results obtained in Section 4.4.

Looking at more than just the work done in this project, for example in TrackMeNot [7] and [6], it is clear that across the board there is a great amount of dependence on the information which Google, and other such companies, provide when it comes to the determination of learning. Whether it is through the interest profiles which the company provides itself, or relying on ads to appear in a very specific manner, one small change is enough to completely invalidate any manner of assessment. Coming up with a robust solution to this problem is crucial to making progress towards giving users full agency over the flow of their personal data. Unfortunately, without fully understanding the inner workings of these systems, this is a difficult task. It is possible that the increased transparency which will come with the GDPR might provide some means of solving this challenge.

Bibliography

[0] E. Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.

[1] L. Sweeney. *Discrimination in online ad delivery*. Queue 11.3 (2013): 10.

[2] Michael Barbaro and Tom Zeller Jr. *A Face Is Exposed for AOL Searcher No. 4417749.* The New York Times. August 9, 2006.

[3] P. Mac Aonghusa and D. J. Leith. *Don't let google know I'm lonely!* arXiv: 1504.08043v2, 2016.

[4] P. Mac Aonghusa and D. J. Leith. *It wasn't me! Plausible Deniability in Web Search*. arXiv:1609.07922, 2016

[5] M. Murugesan and C. Clifton. *Providing privacy through plausibly deniable search*. Proceedings of the 2009 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2009.

[6] M. Degeling and T. Herrmann. *Your interests according to google-a profile-centered analysis for obfuscation of online tracking profiles*. arXiv preprint arXiv:1601.06371 (2016).

[7] D. C. Howe and H. Nissenbaum. *TrackMeNot: Resisting surveillance in web search.* Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society 23 (2009): 417-436.

[8] V. Toubiana, L. Subramanian, and H. Nissenbaum. *Trackmenot: Enhancing the privacy of web search*. arXiv preprint arXiv:1109.4677 (2011).

[9] Do Not Track - Universal Web Tracking Opt Out. [Online] Available: http://donottrack.us/ (Accessed 09 May 2017)

[10] R. Balebako et al. *Measuring the effectiveness of privacy tools for limiting behavioral advertising*. Web, 2012.

[11] Microsoft Technet. *Virtual Private Networking – On Overview*. [Online] Available: https://technet.microsoft.com/en-us/library/bb742566.aspx (Accessed 09 May 2017)

[12] Demon Fiddler. *Random Walk*. [Online] Available: https://addons.mozilla.org/en-US/firefox/addon/random-walk/ (Accessed 27 April 2017)

[13] Matagus. *Google Redirects Fixer & Tracking Remover*. [Online] Available: https://addons.mozilla.org/en-us/firefox/addon/google-no-tracking-url/ (Accessed 27 April 2017)

[14] EU GDPR. *Key Changes with the General Data Protection Regulation.* [Online] Available: http://www.eugdpr.org/the-regulation.html (Accessed 27 April 2017)

[15] Mozilla Developer Network. *What are WebExtensions?* [Online] Available: https://developer.mozilla.org/en-US/Add-ons/WebExtensions/What_are_WebExtensions (Accessed 09 May 2017)

[16] Mocha – JavaScript test framework. [Online] Available: https://mochajs.org/ (Accessed 09 May 2017)

[17] Jasmine Documentation. [Online] Available: https://jasmine.github.io/ (Accessed 09 May 2017)

[18] QUnit – JavaScript unit testing. [Online] Available: https://qunitjs.com/ (Accessed 09 May 2017)

[19] Selenium – Web Browser Automation. [Online] Available: http://www.seleniumhq.org/ (Accessed 09 May 2017)

[20] W3C. *WebDriver*. [Online] Available: https://w3c.github.io/webdriver/webdriver-spec.html (Accessed 09 May 2017)

[21] W3Counter. *Web Browser Market Share Trends.* [Online] Available: https://www.w3counter.com/trends (Accessed 09 May 2017)

[22] StatCounter GlobalStats. *Top 5 Browsers on December 2016.* [Online] Available: http://gs.statcounter.com/#desktop-browser-ww-monthly-201612-201612-bar (Accessed 09 May 2017)

[23] Mozilla Developer Network. *DOMContentLoaded.* [Online] Available: https://developer.mozilla.org/en/docs/Web/Events/DOMContentLoaded (Accessed 09 May 2017)

[24] Mozilla Developer Network. *readystatechange.* [Online] Available: https://developer.mozilla.org/en-US/docs/Web/Events/readystatechange (Accessed 09 May 2017)

[25] Mozilla Developer Network. *load.* [Online] Available: https://developer.mozilla.org/en-US/docs/Web/Events/load (Accessed 09 May 2017)

[26] Mozilla Developer Network. *loadend.* [Online] Available: https://developer.mozilla.org/en-US/docs/Web/Events/loadend (Accessed 09 May 2017)

[27] Mozilla Developer Network. *progress.* [Online] Available: https://developer.mozilla.org/en-US/docs/Web/Events/progress (Accessed 09 May 2017)

[28] Mozilla Developer Network. *Document.readyState.* [Online] Available: https://developer.mozilla.org/en/docs/Web/API/Document/readyState (Accessed 09 May 2017)

[29] D. Le Moal. *I/O Latency Optimization with Polling* in Linux Storage and Filesystems Conference, March 2017. [Online] Available: http://events.linuxfoundation.org/sites/events/files/slides/lemoal-nvme-polling-vault-2017-final_0.pdf (Accessed 11 May 2017)

[30] Monorail. *ChromeDriver Bug Reports.* [Online] Available: https://bugs.chromium.org/p/chromedriver/issues/detail?id=1536&q=selenium&colspec=ID%20Status%20Pri%20Owner%20Summary (Accessed 11 May 2017)

[31] PhantomJS. [Online] Available: http://phantomjs.org/ (Accessed 13 May 2017)