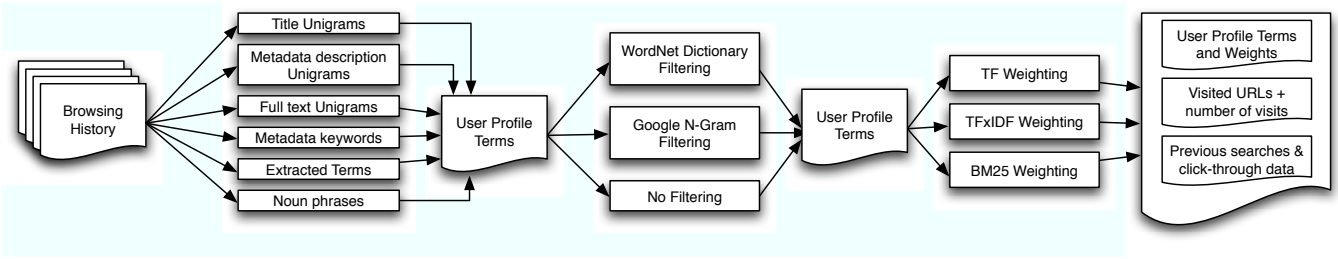

Figure 1: User Profile Generation Steps and Workflow

3. PERSONALIZATION STRATEGIES

In this section, we describe our personalization approach. The first step consists of constructing a user profile, that is then used in a second phase to re-rank search results.

3.1 User Profile Generation

A user is represented by a list of terms and weights associated to those terms, a list of visited URLs and the number of visits to each, and a list of past search queries and pages clicked for these search queries. This profile is generated as shown in Figure 1. First, a user's browsing history is collected and stored as (URL, HTML content) pairs. Next, this browsing history is processed into six different summaries consisting of term lists. Next, the terms can be filtered, and finally term weights are generated using three different weighting algorithms. We now describe each of these steps in detail.

3.1.1 Data Capture

To obtain user browsing histories, a Firefox add-on called AlterEgo[3] was developed. To respect a user's privacy as much as possible, a random unique identifier is generated at installation time. This identifier is used for all data exchange between the add-on and the server recording the data[4]. Participants for this study were recruited via a website explaining the purpose and consequences to potential users, publicized on various e-mail lists, resulting in 50 participants taking part. Whilst we expect that most of these participants are employed in the IT industry due to the recruitment process, a number of people outside of the IT industry without significant web search experience participated as well.

Every time a user with this add-on installed leaves a non-secure (non-https) web page, the add-on transmits the current user's unique identifier, the URL of the page, the time that was spent on the page, the current date and time, and the length of the source HTML to the server. The server then adds the record to a queue of items to process. The server attempts to fetch the source HTML of all pages in the queue. This is performed server-side to ensure that only publicly-visible data is used. Once the source HTML is received, the server compares its length to the length received from AlterEgo. If the length difference is smaller than 50 characters, the HTML is accepted and saved along with the unique identifier, URL, duration and date and time into the database. Otherwise, we assume the content probably came from a password protected but non-secure site (e.g. FaceBook, Hotmail etc.) and the record is discarded.

[3]The source code for this add-on can be downloaded from http://github.com/nicolaasmatthijs/AlterEgo

[4]Note that it is necessary for this data to be collected by our server for research purposes, but our approach does not require this data to be centralized. Our entire method can execute client-side, avoiding the privacy concerns that otherwise arise with server-based approaches.

Table 1: Captured Data Statistics

Metric	Total	Min	Max	Mean
Page Visits	530,334	51	53,459	10,607
Unique Page Visits	218,228	36	26,756	4,364.56
Google Searches	39,838	0	4,203	797
Bing Searches	186	0	53	3.72
Yahoo Searches	87	0	29	1.74
Wikipedia Pages	1,728	0	235	34.56

The add-on captured data for three months from March to May 2010. As shown in Table 1, a total of 530,334 page visits (or an average of 10,607 page visits per user) were recorded. 58% of the visits were to unique pages. The add-on also recorded 39,838 Google searches, 186 Bing searches and 87 Yahoo! searches, indicating that our users are strongly biased towards Google as their search engine, hence Google is used as the baseline in our experiments. An average user issued 797 queries over the three months, indicating that at least 7.5% of all web requests are search related.

3.1.2 Data Extraction

We considered the following summaries of the content viewed by users in building the user profile:

Full Text Unigrams
The body text of each web page, stripped of html tags.

Title Unigrams
The words inside any <*title*> tag on the html pages.

Metadata Description Unigrams
The content inside any <*meta name="description"*> tag on the html pages.

Metadata Keywords Unigrams
The content inside any <*meta name="keywords"*> tag on the html pages.

Extracted Terms
We implemented the Term Extraction algorithm based on Unithood And Termhood Unification as presented in [34], running it on the full text of each visited web page. This algorithm uses the C/NC method, which uses a combination of linguistic and statistical information to score each term. Term candidates are found using a number of linguistic patterns and are assigned a weight based on the frequency of the term and its subterms. This is supplemented with term