ABSTRACT
We propose and motivate a scheme for classifying queries submitted to a people search engine. We specify a number of features for automatically classifying people queries into the proposed classes and examine the effectiveness of these features. Our main finding is that classification is feasible and that using information from past searches, clickouts and news sources is important.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Search process; H.3.3 [Information Storage and Retrieval]: Query formulation

General Terms
Theory, Experimentation, Measurement

Keywords
People search, query log analysis, classification

1. INTRODUCTION
People search is an important aspect of human search behavior. E.g., in web search an estimated 11–17% of the queries contain person names [1]. People search for themselves, people from their past, friends, colleagues, business associates, etc. We examine the transaction logs of a people search engine and are particularly interested in classifying the queries submitted to the engine. What types of queries are submitted? That is, what types of people are searchers looking for? The main aim of the paper is to point out that there are different types of people queries; we propose a taxonomy and describe features for automatically classifying people queries according to this taxonomy.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WebSci ’11, June 14-17, 2011, Koblenz, Germany. Copyright 2011 ACM.
logs used. We refer the reader to [10] for further details on the query information about users is of great value. e.g. personalizing the search interface. In general, however, this setting care should be taken with using the cookies for may occur that one and the same person is represented by make use of the same browser over time, and likewise it tween them. Of course it may occur that several people cookie with a maximum time interval of forty minutes be-

We define sessions as consecutive query entries with the same domain (TLD), a timestamp and a persistent cookie ID.

We address the following research questions:

• Is automatic classification into low and high-profile queries feasible?

• Can we also distinguish event-based and high-profile queries with reasonable accuracy?

• What kind of features are most useful for this task?

2. DATA AND METHODS

The query logs we use were made available by a Dutch lan-
guage people search engine; they contain queries and click-
outs. Query entries consist of a first name, last name, an optional keyword, a timestamp and an associated unique ID of a persistent cookie. Even though a cookie need not corre-
respond one to one with a person, we interpret it as a unique visitor, i.e., a user. Clickouts consist of a URL, a top level domain (TLD), a timestamp and a persistent cookie ID.

We refer the reader to [10] for further details on the query logs used.

Annotation. We manually annotated instances of queries that are issued by at least twenty different users over time from September 1st until December the 31st. We then sam-
ples only instances from October 1st onwards, to give us for each query at least a month history to extract features from. This does not imply that there are no low profile query instances. First, if the query instance is one of the first searches for this query, it may well be annotated as low pro-
file: we noted already that the class of a query may change over time. Also, names are ambiguous, and searches for vari-
ous low profile persons with the same name may add up to above the threshold.

Annotators were asked to determine if a person is high-
profile or not (i.e., low-profile) at the time of the query in-
stance. In case an annotator labeled a target as high-profile, we also required a decision whether the query instance was event-based high-profile or regular high-profile. This deci-
sion can be subtle; we illustrate this with an example.

In Figure 1 we plot the search volume and the number of mentions in RSS feeds of national newspapers of two high-
profile targets, highlighting the difference between a ‘regu-
lar’ and an ‘event-based’ instance. On the left-hand side, we show the graph of a controversial politician frequently mentioned in the news (Geert Wilders). The query instance is most likely not related to a particular event, but rather to the sum of the events that made him well-known, it is a regular high profile instance. On the right-hand side we display the search volume and number of mentions in RSS feeds of an actress who is much less mentioned in the news; however, when she is mentioned in the news (because of the tragic passing of her husband), this is followed by a very clear peak in search volume. This instance is a clear exam-
ple of an event-based high-profile query.

Figure 1: Search volume (black lines) and mentions in RSS feeds of national newspapers of two high-profile targets, highlighting the difference between a ‘regular’ and an ‘event-based’ instance.
Classification. For automatic query classification, we use the features listed in Table 1. There are six groups of features, with 16 features in total: search volume, click volume, news volume, Wikipedia presence, burstiness and clickout entropy.

We used three standard classifiers: a J48 decision tree classifier, a Naïve Bayes classifier (NB) and a support vector machine (SVM) to classify the instances: we used the implementations available in the Weka toolkit [3]. The SVM performance on which we report below is obtained with a cost parameter of 1 and a linear kernel, without feature normalization. We report on precision (P) and recall (R) per class for a stratified ten fold cross validation experiment.

3. RESULTS AND DISCUSSION
We report on two experiments: (i) a two-way experiment in which we aim to automatically distinguish between high-profile and low-profile people queries and (ii) a three-way experiment in which we aim to distinguish between event-based high-profile, regular high-profile and low-profile queries. The results of both classification experiments are given in Table 2 below.

After discussing the outcomes of the two experiments we will analyze the results of the J48 algorithm in more detail because (i) it is the best overall performing classifier in our experiments and (ii) because it produces models that are easily interpretable.

In our setting decision tree classifiers like J48 perform well because they can combine nominal and ratio features and they handle dependencies in features well. Our features are somewhat redundant and depend on each other, e.g., if the average unique visitors per day that entered a given query instance is high, the average over the week before the query is more likely to be high. Since Naïve Bayes assumes class conditional independence of features, this may explain why it performs a bit less.

3.1 High profile versus low profile classification
We first examine the outcomes of the two-way classification experiment; see the top half of Table 2. Clearly, it is feasible to classify query instances into the high- and low-profile classes with a C4.5 decision tree classifier. Recall of the high-profile instances is a bit worse with Naïve Bayes and an SVM.

In Figure 2 we show a partial decision tree. This tree is learned on the entire dataset. On each vertex the training samples are split on the indicated feature, see Table 1. Each edge shows the threshold value on which it is split. The leaf nodes indicate the class that the tree will predict for new examples that satisfy the requirements to reach the node. "H" and "L" represent the high- and low-profile classes. Between brackets the number of queries within that class is listed. If training examples are misclassified their number is reported after a slash. Some leaf nodes contain a feature and a number of classes. Here the decision tree visualization was truncated to save space; the feature listed will yield the next splitting criterion; the classes show in parentheses how many nodes indicate the class that the tree will predict for new samples that satisfy the requirements to reach the node. "H" and "L" represent the high- and low-profile classes. Between brackets the number of queries within that class is listed. If training examples are misclassified their number is reported after a slash. Some leaf nodes contain a feature and a number of classes. Here the decision tree visualization was truncated to save space; the feature listed will yield the next splitting criterion; the classes show in parentheses how many

<table>
<thead>
<tr>
<th>Name (abbr)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search volume</td>
<td>in average unique daily visitors per day</td>
</tr>
<tr>
<td>– three months (SVFS)</td>
<td>from Sep 1st - date of this query instance</td>
</tr>
<tr>
<td>– last week (SV7D)</td>
<td>last week before this query instance</td>
</tr>
<tr>
<td>– trend (SVT)</td>
<td>difference between the previous two: SV7D – SV3M</td>
</tr>
<tr>
<td>Click volume</td>
<td>in average clicks per day over</td>
</tr>
<tr>
<td>– three months (CVFS)</td>
<td>from Sep 1st - date of this query instance</td>
</tr>
<tr>
<td>– last week (CV7D)</td>
<td>last week before this query instance</td>
</tr>
<tr>
<td>– trend (CVT)</td>
<td>difference between the previous two: CV7D – CV3M</td>
</tr>
<tr>
<td>News volume</td>
<td>in average mentions in RSS feeds of national news papers per day</td>
</tr>
<tr>
<td>– three months (NVFS)</td>
<td>the Sep 1st–date of this query instance</td>
</tr>
<tr>
<td>– last week (NV7D)</td>
<td>the last week before this query instance</td>
</tr>
<tr>
<td>– trend (NVT)</td>
<td>difference between the previous two: NV7D – NV3M</td>
</tr>
<tr>
<td>Wikipedia presence</td>
<td>calculated with Dutch Wikipedia dump dated August 26, 2010</td>
</tr>
<tr>
<td>– title match (WPTM)</td>
<td>query person name matches title Wikipedia page (yes or no)</td>
</tr>
<tr>
<td>– frequency (WPF)</td>
<td>frequency of occurrence of person name in Wikipedia</td>
</tr>
<tr>
<td>Burstiness</td>
<td>where a burst is a peak in the search volume history of this query: consecutive days with volume at least two standard deviations above the mean [7]</td>
</tr>
<tr>
<td>– number of bursts (NB)</td>
<td>ratio search volume in bursts and total search volume (BV/SV)</td>
</tr>
<tr>
<td>– one over the number of days since last burst (1/DsLB)</td>
<td></td>
</tr>
<tr>
<td>Clickout entropy</td>
<td>as defined in [7]: – ∑_{d∈D} P(c_d) * log_2 P(c_d), where P(c_d) is the probability of click on d ∈ D, and D is:</td>
</tr>
<tr>
<td>– the set of unique urls (CEU)</td>
<td></td>
</tr>
<tr>
<td>– the set of unique top level domains (CETLD)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Features grouped by type. There are six types of features, and 16 features in total.

of the training examples are subsequently (mis)classified.

The most important feature is the average number of clicks
Table 2: Results of two stratified ten fold cross validation experiments.

<table>
<thead>
<tr>
<th>Query type</th>
<th>C4.5</th>
<th>NB</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>High-profile</td>
<td>0.85</td>
<td>0.82</td>
<td>0.89</td>
</tr>
<tr>
<td>Low-profile</td>
<td>0.89</td>
<td>0.91</td>
<td>0.81</td>
</tr>
<tr>
<td>Event-based</td>
<td>0.83</td>
<td>0.87</td>
<td>0.74</td>
</tr>
<tr>
<td>Regular</td>
<td>0.57</td>
<td>0.54</td>
<td>0.53</td>
</tr>
<tr>
<td>Low-profile</td>
<td>0.92</td>
<td>0.90</td>
<td>0.81</td>
</tr>
</tbody>
</table>

Figure 2: Partial decision tree for the two way classification experiment.

Per day over the last week. A surprisingly low number of clicks is sufficient to classify as many as 32 queries as high profile queries. One explanation for this is that clicks in the people search engine we study require substantial effort on the part of the user. Search results are displayed grouped by search engine or social media platform. If a user wants to find e.g., a social media profile, she has to expand the results for the social media platform of choice, then a text snippet (a short description of a particular search result) is displayed and an outlink may be followed. This explains why there are not many clicks in the query log files. A few clicks may well have resulted from many searches.

The second feature used is the number of bursts. This feature uses the search volume history. If there are one or more bursts, then the query is high profile. In the absence of bursts, the third split is counterintuitive. The average unique number of unique visitors on which the remaining set is split seems very low. Even so, queries that were issued by even fewer people are all high profile queries in this dataset. And the bulk of the instances with a higher search volume is low-profile. This is surprising because we defined high profile persons as well-known people, either because of some recent event (event-based) or because they are a public figure, celebrity, or generally much sought after.

The news volume features do not appear at all in the decision tree for the two-way experiment. We will see that they do play a role in the three way experiment, however.

3.2 Low profile, event based and regular high profile classification

We now turn to the three-way classification experiment; see the bottom half of Table 2. Three-way classification into event-based high-profile (“H”), regular high-profile (“R”) and low-profile (“L”) is harder than two-way classification. For J48, performance on the low-profile and event-based high-profile is reasonable, but precision and recall for regular high-profile needs improvement. Results for this category suffer from the fact that there are only 24 regular high-profile instances in the data set. Looking at the Naive Bayes and SVM results, mainly recall for the high-profile classes is lower compared to J48.

Figure 3: Partial decision tree for the three way classification experiment.

We can learn the contribution of individual features from the learned decision tree on the entire dataset in Figure 3. The first feature is again the average number of clicks per day over the last week before the query. But this time if it is higher than 2.0 the news volume comes into play. It seems counterintuitive that a low average number of mentions in the news per day over the last week leads the classifier to the conclusion that the query instance is event based. However: a few mentions in the news are often enough to cause a large interest in the person. If somebody passes away, this is may be followed by a peak in search volume in the people search logs even if it hardly mentioned in the news. Many mentions in the news can be a sign that a person is famous but not well-known because of a particular event.

Again, the number of bursts is an important feature. In the absence of bursts, we find many low profile queries. Again
there is the curious exception of searches that also have a low average search volume until the date of the current instance: these are all event based queries. If there are bursts we see again that regular high profile queries have a higher search volume.

3.3 Lessons learned in the two experiments

The similarities between the decision trees for both experiments are clear: the click and search volume features appear with the same threshold values. This is not very surprising as the high profile class is nothing more than the union of the event based and the regular high-profile class. There are also differences. When high-profile searches have to be split into event based and regular query instances, the news volume feature group is one of the top features. Moreover, a Wikipedia feature appears. From each group in Table 1 a feature is now being used, except for the clickout entropy features: evidence from clicks, searches, news sources and Wikipedia all contribute.

Another finding is that different features from the same groups are quite redundant. From each group typically only one feature plays a prominent role in the decision trees.

We can now answer our research questions posed in the introduction.

- **Is automatic classification into low and high-profile queries feasible?**

  Performance of the decision tree classifier was very high in terms of recall and precision. Therefore it is feasible.

- **Can we also distinguish event-based and high-profile queries with reasonable accuracy?**

  No, not quite. The precision and recall values for regular high-profile queries are low. There may be several causes for this. First, there were only 24 regular high profile queries in the dataset. Second, more features may need to be added. Particularly promising may be features obtained from the document collections being searched.

- **What kind of features are most useful for our classification tasks?**

  Features that use clickouts, search volume and news volume are all important, especially for the three way task. It is not very useful to add much redundancy. For example, none of the “trend” features from Table 1 appeared high in the two decision trees. The same holds for the clickout entropy features.

4. CONCLUSIONS

We proposed a query classification scheme for a specific vertical search engine, viz. a people search engine. The scheme consists of low-profile people queries, event-based high-profile queries and regular high-profile queries. We have shown that people query instances can be automatically classified into high-profile queries and low-profile queries with high precision and recall scores. Features that appeared to be particularly informative are click volume and the number of bursts. A further three-way classification into event-based and regular high-profile queries is harder. Here, the most informative features use clickouts, search volume and number of mentions in the news.

In future work we plan to examine the use of the type of clickouts, such as clickouts to social media, other search engines, and news sites, as well as features derived from the document collections being searched.

5. ACKNOWLEDGEMENTS

This research was supported by the European Union’s ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, the PROMISE Network of Excellence co-funded by the 7th Framework Programme of the European Commission, grant agreement no. 258191, the DuOMAN project carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments under project nr STE-09-12, the Netherlands Organisation for Scientific Research (NWO) under project nr 612.061.814, 612.061.815, 640.004.802, 380-70-011, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the WAHSP project funded by the CLARIN-nl program, and the COMMIT project “Information Retrieval for Information Services.”

6. REFERENCES