ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/263230454

Query intent detection based on query log mining

Article *in* Journal of Web Engineering (JWE) · March 2014

CITATIONS
5

READS
1,523

3 authors, including:

Juan Zamora
Pontificia Universidad Católica de Valparaíso
15 PUBLICATIONS **69** CITATIONS
SEE PROFILE

Marcelo Mendoza
Universidad Técnica Federico Santa María
112 PUBLICATIONS **4,693** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Measuring diversity in complex systems View project

Project Implementation of Tools and Processes to the Chilean Virtual Observatory View project

All content following this page was uploaded by Marcelo Mendoza on 19 June 2014.

The user has requested enhancement of the downloaded file.

Journal of Web Engineering, Vol. 13, No. 1&2 (2014) 024–052
© Rinton Press

QUERY INTENT DETECTION BASED ON QUERY LOG MINING

JUAN ZAMORA, MARCELO MENDOZA, and HÉCTOR ALLENDE
Computer Science Department, Universidad Técnica Federico Santa María
Av. España 1680, Valparaíso, Chile
{juan.zamora,marcelo.mendoza,hector.allende}@usm.cl

Received August 22, 2012
Revised March 12, 2013

In this paper we deal with the problem of automatic detection of query intent in search engines. We studied features that have shown good performance in the state-of-the-art, combined with novel features extracted from click-through data. We show that the combination of these features gives good precision results. In a second stage, four text-based classifiers were studied to test the usefulness of text-based features. With a low rate of false positives (less than 10 %) the proposed classifiers can detect query intent in over 90% of the evaluation instances. However due to a notorious unbalance in the classes, the proposed classifiers show poor results to detect **transactional** intents. We address this problem by including a cost sensitive learning strategy, allowing to solve the skewed data distribution. Finally, we explore the use of classifier ensembles which allow to us to achieve the best performance for the task.

Keywords: Query categorization, user intents, query logs

Communicated by: D. Schwabe & F. Vitali

1 Introduction

The Web is the biggest and most dynamic database of documents in the world: made up by billions of pages and Websites of diverse content and quality, therefore searching for relevant information becomes increasingly difficult. To face this task, Web users often use search engines which provide a simple interface to formulate queries. After processing the query, the search engine determines a collection of recommended pages/sites displayed in order of relevance, using a ranking function.

The user's queries are typically short and ambiguous, frequently made up of two or three imprecise terms. As a result, the list of pages/sites can be extremely long and/or only marginally relevant.

The limitation of text-based information retrieval methods in the Web domain is due fundamentally to that not all user needs are related to information searches. The user's purpose is often to use the Web to support a transaction, or find a particular site from which to navigate to different related sites. In these cases, the use of a classic text-based information retrieval method can limit the effectiveness of the search engines.

Taxonomies of Web searches

Andrei Broder [4] posed that the needs behind Web searches were not only informational, but rather **navigational** and **transactional** as well. To show the prevalence of his taxonomy, Broder utilized two methods: A survey of Altavista users and an analysis of a query log at Altavista. The survey basically asks about three features of the query: Specific site in mind, kind of desired interaction and dispersion level of desired results. Using 3,190 valid user returns, he estimated a proportion of 40%, 35% and 25% of informational, **transactional** and **navigational** queries, respectively. The second method used by Broder was a query log analysis: he inspected 400 queries and estimated a proportion of 50%, 30% and 20% of informational, **transactional** and **navigational** queries, respectively.

Following the research line posed by Broder, Rose and Levinson [26] tried to answer why users perform searches and proposed a framework for understanding the underlying goals of user searches. They refined Broder's taxonomy by organizing a set of canonical goal categories within it. Later, Baeza-Yates et. al [2] proposed a new taxonomy based on Broder's, but also considering ambiguous user intent. Additionally, they proposed a query vector model based on text and click-through data to classify user intent in the given taxonomy.

Considering every one of the three categories proposed by Broder, it would be expected that search engines should be capable of identifying user intent. By capturing the user intent behind Web queries, a search engine should rank higher those pages/sites that best satisfy the user's need.

Motivation: How query intent detection can help search quality

The purpose of query intent detection is to help search quality. Text-based retrieval methods can recover meaningful results if query terms and document terms match. At some extent, **navigational** and **transactional** intents cannot be solely expressed by the detection of coincidences between document and query terms. Thus, link-based or click-based features may be required to create better ranking functions. We illustrate this fact by conducting an experiment over the well known dataset LETOR 3.0 [19]. LETOR contains a number of collections of query - document pairs, among them the TREC Homepage finding challenge (from here on called HP), where the user requires to find a specific site/page, and the TREC Topic distillation challenge (from here on TD), where the user requires information related to a particular topic. Notice that HP and TD are **navigational** and informational intents. A summary of each dataset used in this section is provided in Table 1.

Dataset	Year	Qus.	Docs.	Pairs $q-d$	Rels.
H.P.	2003	150	123,521	147,605	183
H.P.	2004	75	69,059	74,408	81
T.D.	2003	50	47,239	49,057	407
T.D.	2004	75	70,096	74,145	1,116

Table 1: Characteristics of the LETOR's datasets used in this section.

The third column indicates the number of queries that the dataset considers, the fourth the number of documents, the fifth the number of query-document pairs evaluated and of these, in the sixth column, how many of them have been considered relevant by the experts of TREC.

We evaluate search quality under two different scenarios. In a first one, a global ranking function is used for both collections. In a second one, a local ranking function is used for HP ranking and another one for TD ranking. Then we compare both situations in terms of search quality.

Genetic algorithms were used to generate the ranking functions. This method was explored previously by Fan *et al.* [9]. We considered this technique because allow us to identify a specific ranking function, offering evidence for analysis purposes. Statistical learning methods as support vector machines achieve better quality results but making difficult feature analysis, reason why we discarded its use in this first experiment.

For the learning process we have considered features based on content, hyperlinks and hybrid measures. A summary of the considered features can be found in Table 2. In total, we considered 64 features. The last column shows how many features of each type contain LETOR 3.0. The content features were calculated on the title, body, URL, anchor and whole document. The features that were evaluated in the extracted title were considered for BM25 and for the three variants of LMIR (language models-based features). For the features based on hyperlinks, HITS considered the hub and authority scores separately. The same fact happeeds when Topical HITS was calculated. As for the hybrid features, Site Propagation considered the variants score and term. Link propagation considered two variants, score and feature, each one calculated for incoming links and outgoing links (in the last case the normalized out variant was considered as well).

Type	Feature	Num. of Feat.
Content	Tf	5
	Idf	5
	Tf-Idf	5
	DL	5
	BM25	5
	LMIR.abs	5
	LMIR.dir	5
	LMIR.jm	5
	Ext. title	4
Hyperlink	HITS	2
	PageRank	1
	HostRank	1
	Top. HITS	2
	Top. PR	1
	Inc. links	1
	Out. links	1
Hybrid	Site prop.	2
	Link prop.	6
	Num. slash	1
	Num. child	1
	URL length	1

Table 2: Features considered for the ranking function learning process.

Five folds cross-validation was used for performance evaluation. As a fitness function the F-measure was used. For each dataset, the training and validation phase have been implemented measuring the results obtained in average fitness over the five partitions. Table 3 lists these results.

Dataset	Training	Validation	Time elapsed [s]
2003 HP	0.629	0.604	47,756
2003 TD	0.278	0.178	28,370
Weighted HP + TD	0.541	0.497	-
Global 2003	0.320	0.342	72,574
2004 HP	0.504	0.400	47,162
2004 TD	0.248	0.219	46,819
Weighted HP + TD	0.376	0.310	-
Global 2004	0.194	0.166	95,901

Table 3: Global performance results of the learning the ranking function process.

The global ranking function results are listed in the rows indicated by Global 2003 and Global 2004. The rows indicated by Weighted HP + TD were calculated from the weighted average of HP and TD regarding the number of queries of each collection. The gap between weighted and global (around 20 and 15 percentual points for training and validation in both datasets, respectively) indicates the gain that the use of two different ranking functions shows against the use of only one global ranking function. These results illustrates that search quality can be affected from 14% to 22% in F-measure if a global ranking function is used for these queries.

We illustrate which features have been most meaningful for each dataset. For each of these ranking functions, the number of variables of each type (content, hyperlink, hybrid) has been counted. Then, the proportion of variables of each type, across the five folds was calculated. Table 4 shows these results. Bold fonts indicate prevalent features for each dataset.

Dataset	Content	Hyperlink	Hybrid
2003 HP	36.05	10.47	**53.48**
2003 TD	**64.53**	14.29	21.18
2003	33.33	25.00	**41.67**
2004 HP	**69.79**	12.50	17.71
2004 TD	**58.37**	13.06	28.57
2004	**59.83**	17.09	23.08

Table 4: Feature distribution for each dataset.

Table 4 shows that the distribution of the most used features varies depending on the dataset considered. The features most used in TREC 2004 are based on content, independent of the type of search done. The results are different for TREC 2003, where the content features are more relevant for TD searches and the hyperlink and hybrid features are more relevant for HP searches. In general, TREC 2003 requires more hybrid features, however the proportion of this type of characteristic grows significantly when considering only HP, illustrating that the design of specific ranking functions for different types of query intents can be meaningful for search quality.

Problem statement

A number of articles have tackled the problem of automatically identifying user intents in search engines as we will show in the state-of-the-art section. Mainly two lines of work have received the focus of research. The first and more active one attempts to build classification algorithms based on features extracted directly from the query such as the number of terms in it, or rather information about the interaction of the user with the search engine (e.g. number of pages/sites selected in the session). The second line of research is centered on establishing connections between the query and selected documents from it. The latter is often achived by constructing term vector representations of queries (or of the sessions).

In general, the methods that we will discuss in the state-of-the-art show that it is possible to automatically identifying the user intent, being **transactional** goals the most difficult to predict. However this problem is far from being resolved mainly because an evaluation of all of these methods on a single dataset is lacking, thereby preventing comparative evaluations to be done on them. Also, in general, the methods of the state-of-the-art have been evaluated on few queries, which better corresponds to exploratory studies. Finally, comparative analyses at feature level are lacking. Studies determining for example the relationship of dependency among features, or the proposition of strategies that allow them to be combined adequately to improve the performance of the identification are necessary. We cope with these challenges.

In this paper, we describe the process of creating a new dataset that includes more queries and more features that any of the ones previously used in the state-of-the-art, releasing it to the public for research purposes. We develope a descriptive and comparative analysis of the features for the dataset, establishing possible relationships of dependency among them. Later, we evaluated the behavior of the features most commonly used in the state-of-the-art constructing a rule based classifier. Then, we propose new classifiers that operates on the vector space of query terms and documents, combining information from text and query logs in a new way. Experimental results show the feasibility of our approach but with some flaws. First, due to the data imbalance, **transactional** intents tend to be labeled as **navigational** or informational. We address this problem by performing cost sensitive learning, which allow to solve the skewed nature of the training data. Finally, we explore the combination of the classifiers to get better search quality results. Our experiments show that the use of ensembles allow to really solve the query intent detection task.

Contributions

This article is a substantially improved version of our earlier work [21, 22]. Most of the sections have been reviewed and completed. In addition, this version describes the creation of a new dataset for query intent evaluation. This dataset has more queries and features than any other available on-line, allowing a more consistent performance of the comparison between the proposed methods and future works. Next, a feature analysis is made to gain understanding and give support to the proposed methods. Furthermore, this version presents the use of costs sensitiveness for learning **transactional** intents which show significant improvements in search quality. In addition, we present a study of classifier combinations, that effectively allow to get better query intent detection results.

2 Related work

The state of the art shows that several articles have tackled the problem of automatically identifying user's intentions. Considering the categories established in Broder's taxonomy, they have faced the problem proposing features of the queries and/or query sessions from those that the user's intention can be distinguished. In this sense, Kang and Kim [16] introduced a query classifier that is able to identify informational and navigational intentions. To do this, they separated a collection of documents in two sub collections, one for navigational pages/sites and the other for informational ones. To separate the collection, each page/site that corresponded to a *root* of a specific site was added to the navigational collection, otherwise it was classified as informational. Then, they constructed a language model for each collection. Using queries extracted from the collection TREC 2001 that corresponds to types of searches *homepage finding* or *topic finding* they studied the behavior of different measures of relevance between the terms that make up the queries and the terms that describe each document collection. Among the measures studied *distribution difference* and *mutual information* stand out. They showed the connection of the occurrence of the query terms in each collection. Using 200 queries from TREC to evaluate the classifier, they obtained 91.7% precision and 61.5% recall.

Lee *et al.* [17] tried to determine the feasibility of automatic detection of user's intention. For this, the authors subjected 28 students from the Computer Science Department at UCLA to the evaluation of the type of intention that reflected the 50 most popular queries formulated by UCLA in Google. The survey was designed in such a way that to make it possible to identify if the intention of the user was informational or navigational. Approximately 40% of the queries could not reach a consensus so they were categorized as unpredictable. The authors observed that these queries refer mainly to entities (people or software) where it is common to observe divergence in needs (for example, the query *cygwin* which refers to software, could indicate the intention to download the software and also find the homepage). In the same paper, Lee *et al.* studied the distribution of clicks and anchor text analyzing the query logs of queries classified as informational or navigational. They observed that the informational queries presented distributions with heavier tails, which is why measures such as the average, median, *skewness* or *kurtosis* of these distributions could be useful for automatic classification according to query intention. The evaluation of 60% of queries classified as navigational or informational by users shows when these features are combined adequately it is possible to obtain a categorizer with approximately 90% precision.

Kang [15] faced the problem of detecting transactional intentions. He explored information retrieved from hyperlinks and anchor texts. Kang proposed identification of which type of resource corresponded each referenced URL from a given page (site, sub site, music, image, text, application, service, html or file). To do this, he recovered the extensions of the URLs referenced (for example, if the URL ends with *MP3* its type is music). Then, for each of the URLs the anchor text that referenced it were recovered. The anchor text were processed in order to identify *cue expressions*, meaning short expressions that allocate the use that it gives to the resource (for example, *download file* indicates that the resource referenced is a downloadable file). Using this information, Kang introduced a new collection of features called *link scores*, which are calculated as follows. Given the terms of a query, the candidates are extracted *cue expression* (Kang proposed extracting bi-terms besides the first and last term of the query) later calculating the co-occurrences of these expressions in the collection of *cue expressions* that represent each type of resource. In this way, each query obtained nine link scores. These features were used to train a learning machine based on examples (TiMBL [7]). Using queries from TREC 2000 and 2001 a dataset was made with 495 queries (Kang assumed that *homepage finding* is equivalent to navigational intention and *topic relevance* equivalent to informational). The transactional queries from the dataset (100 in all) were extracted manually from a query log from Lycos (lycos.com). The best classifier was the one that combined all of the features considered, reaching a precision of 78% in the detection of transactional intents.

Liu *et al.* [20] proposed using features based on click-through data to distinguish between informational and navigational intentions. Analyzing a query log of more than 80 million queries, they observed that only 16.2% of these queries were able to recover relevant results using anchor texts. Then, they decided to explore new variables based on click-through data allowing along with the previous variable to improve the coverage of the methods. Two new features were proposed: given a query, they proposed calculating the number of satisfied clicks (**n**CS), meaning the fraction of sessions that register less than n clicks, and number of satisfied results (**n**RS) which correspond to the fraction of sessions that register clicks only in the top-n results. Liu *et al.* proposed also using the click features *distribution* introduced by Lee *et al.*, combining them in a decision tree. Using a dataset composed of 153 navigational queries and 45 informational queries tagged manually, a decision tree using the algorithm C 4.5 was trained. Then, using a dataset of 81 informational queries and 152 navigational queries they evaluated the performance of the decision tree reaching 81.49% precision and 81.54% recall. The paper illustrates that in general navigational queries have greater values in **2**CS and **5**RS than informational queries.

Baeza-Yates *et al.* [2] proposed using the terms from snippets of the pages/sites selected in the query sessions to construct vector representations of them. To do this, they used the representation Tf-Pop, introduced previously in [3] in the context of query clustering and applied also to the automatic maintenance of Web directories [11]. Using a dataset of 6,042 manually tagged queries according to informational, non-informational or ambiguous intentions, they constructed vector representations from a query log of the Chilean search engine TodoCl. Then, they trained a classifier based on support vector machines (SVMs), achieving a precision close to 80% with recall above 80% for informational intents. They obtained lower yields in the non-informational case (approximately 60% precision with 40% recall). The paper also shows that in the case of ambiguous queries the prediction is very difficult. In this case the experiments show a precision less than 40% with recall lower than 20%.

Yuan *et al.* [8] studied features based on click-through data and anchor text to distinguish between informational and navigational intents. To do this, they brought in four new measurements based on entropy. *Click entropy* measured the degree of dispersion in the anchor text that produce matches in a given query. In the case of the *click entropy* measurement, it is possible to calculate the dispersion considering the clicks produced in a given domain as coming from the same source, calling this measurement *domain click entropy.* Something similar was done for the distribution of links, considering that anchor texts were produced by only one source from the same site, calling this measurement *site entropy.* Using 206 queries manually categorized as informational and navigational, the authors trained a support vector machine (SVM) using five-fold cross validation, with four folds for training and one for evaluation. The process also considers the measurements nCS and nRS proposed by Liu *et al.* and the medians of the distributions of anchor text and clicks as studied by Lee *et al.*. The best combination of variables corresponds to *site entropy* and *median click*, obtaining a performance close to 97% precision.

Jansen *et al.* [12] proposed the creation of a single classifier to distinguish between informational, navigational and transactional intents. To do this, they characterized each intent according to if-then rules defined by the authors. Among the rules proposed for each intent they emphasized the length of the query (for example, they assumed that a navigational query has at least three terms) and the number of results reviewed by the users (for example, they assumed that a navigational query only registers clicks on the first page of results). They also suggested lists of key terms used frequently in the formulation of transactional and informational queries (for example *download*, *games*, *buy*, *chat* for transactional intentions, *ways to*, *how to*, *list*, *playlist* for informational). Using these lists, they consolidated a database of key terms for informational and transactional intentions. Based on these rules, they classified a million and a half queries taken from Altavista query logs (altavista.com), Excite (excite.com) and AlltheWeb (alltheWeb.com). The classifier showed that the queries were distributed as 80.6% informational, 10.2% navigational and 9.2% transactional. To evaluate the precision of the rules, the authors manually tagged 400 queries taken out of a query log from Dogpile (dogpile.com), attaining 74% precision. The analysis of the experiment results showed over classification of informational intentions (approximately 20% false positives in the whole set) and sub classification of transactional and informational intentions (approximately 6% false positives in the whole set).

The impact of query intent detection has been explored in different domains. For example, Ashkan *et al.* [1] explore the use of query intent detection in the domain of internet monetization strategies. By developing a methodology based on query information extraction and ad clickthrough log mining the extend the taxonomy of Broder, trying to detect commercial intents, that at some extent is closely related to transactional intents. The specific problem of information needs detection is addressed by Radlinski *et al.* [25]. They hypothesize that it is possible to get better query descriptions by inferring topical needs which can be detected by identifying query reformulations in query logs. Finally, Calderón-Benavides *et al.* [5] explore the multidimensional nature of query intents, characterizing a wide range of new query facets, outlining dependencies between these facets. Among the explored facets there are novel aspects related to spatial-temporal dimensions of the intent, as also a facet related to expectation of trusted answers retrieval.

3 Dataset Creation

In this paper we will use a query log from AOL, which has been previously used in other papers as it provides a good sample of the types of searches made by users on the Web [23]. This query log registers more than 20M query instances. We processed the query log containing queries formulated between March and May of 2006, retrieving a random sample equals to the 10% of the original query collection, achieving around 2M query instances related to $51,755$ users and $1,117,000$ clicks over $384,000$ different pages/sites. Then, this sample was processed taking the line-by-line information and exporting it to a relational database. From the available data we retrieved query sessions. According to Wen et al. [28], a query session is compounded by a unique query formulated by a single user in a given point of time and it includes the list of URLs where the user clicked on. By disregarding query sessions without selections, we obtained 759, 000 query sessions, so each user registered an average of 14.6 query sessions.

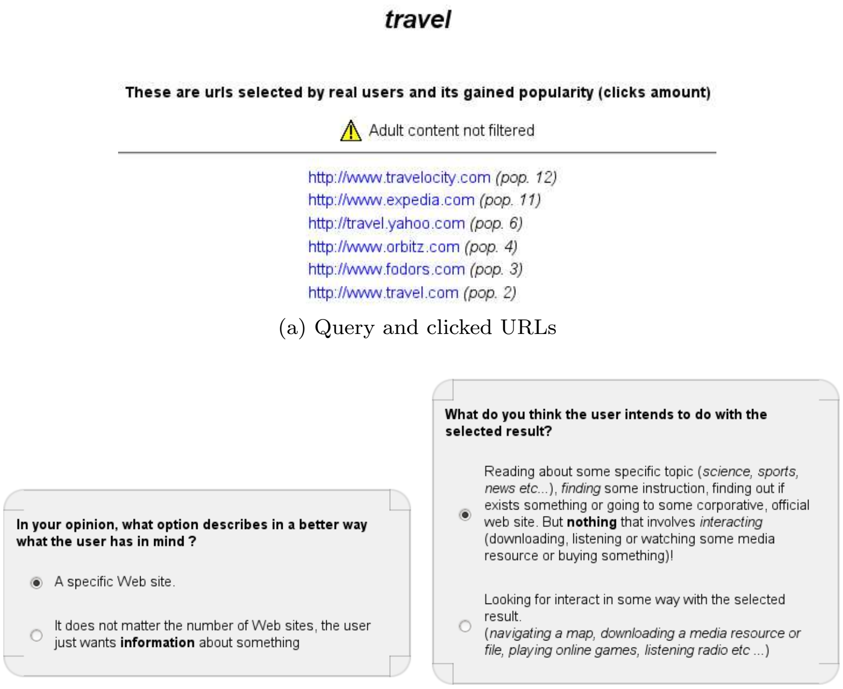
We take advantage of the query session definition in many ways. First, it favorates the recovery of query sessions from the relational database by using a *group by* query operator. Second, it allows to aggregate selections made by different users, favoring the creation of query vector representations related to many users. Finally, it allows to assume that the query intent remains the same for a great proportion of the queries, because each session is related to only one query. This assumption is true if the query is unambiguous. On the other hand, the aggregation of query sessions at query level introduces a challenge, because a fraction of the queries may registers different intents due to polysemy. However, as we will explain in the following section, we will address this problem by elimination of ambiguous queries, according to the measurement of intent agreement among the evaluators of our collection.

3.1 Manual Classification of Queries

Using the processed data, we took a random sample of 2, 000 queries, each one related to at least 10 query sessions. These queries were tagged according to a methodology similar to Broders [4]. The classification methodology consisted on the evaluation of the 2,000 queries by a group of experts, made up of three experts from our laboratories. Each had to answer a questionnaire made up by two questions as depicted in figures 1b and 1c. The text of the query and the history of the selection of results existing in the query log were used as support for the decision (Figure 1a).

As can be observed in figures 1b and 1c, the first question aims to determine if the results sought through the query are concentrated in just one site, usually denoting it as **navigational** intent or if they can be dispersed in several sites, denoting an informational intent. A **transactional** goal could coincide with both options (for example, the queries "*Buy magazine on amazon.com*" or "*Listen online radio*"), for which it is important to ask a second question. The second question seeks to distinguish between the *informational-**transactional*** and the ***navigational-transactional*** pairs. This is done asking about the action that the user could take with the selected result. Reading or taking information from a site would be considered a passive action, while downloading a file, navigating a map, or buying something would be active actions, the latter related with **transactional** intent. Using the answers to questions 1 and 2, the intent is determined based on a set of rules; the set of rules utilized is shown in Figure 2.

The definitive collection of queries used for analysis was that in which the responses from all the experts came to a consensus. Those that remained classified in two or more categories were reevaluated by the group of experts, who took a second instance to reach a consensus about the intent of the query. Finally, queries where no consensus was reached were eliminated from the data collection. As a result, 1, 953 queries were tagged, dismissing only 2% of the initial collection. The proportions were distributed from greatest to least in the informational, **navigational** and **transactional** categories, with 52%, 33%, and 15% respectively.



travel

These are urls selected by real users and its gained popularity (clicks amount)

⚠ Adult content not filtered

http://www.travelocity.com *(pop. 12)*
http://www.expedia.com *(pop. 11)*
http://travel.yahoo.com *(pop. 6)*
http://www.orbitz.com *(pop. 4)*
http://www.fodors.com *(pop. 3)*
http://www.travel.com *(pop. 2)*

(a) Query and clicked URLs

In your opinion, what option describes in a better way what the user has in mind ?

● A specific Web site.

○ It does not matter the number of Web sites, the user just wants **information** about something

What do you think the user intends to do with the selected result?

● Reading about some specific topic (*science, sports, news etc...*), *finding* some instruction, finding out if exists something or going to some corporative, official web site. But **nothing** that involves *interacting* (downloading, listening or watching some media resource or buying something)!

○ Looking for interact in some way with the selected result.
(*navigating a map, downloading a media resource or file, playing online games, listening radio etc ...*)

(b) First question

(c) Second question

Fig. 1: User intents survey

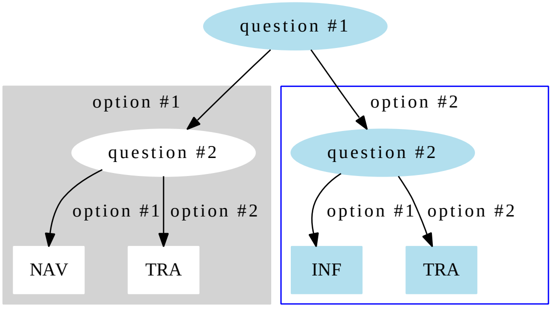


Fig. 2: Rules scheme applied to the user intents survey

We released the dataset for research purposes. For more details please visit the following URL: `http://octopus.inf.utfsm.cl/~juan/datasets/`

3.2 Feature Extraction

Once the query log was processed, the next task was the extraction of some query features successfully used in previous works and also not excessively expensive to compute. We decided to use the features ***n***CS, ***n***RS and Median of Clicks. We also considered the number of terms making up the query as well as studied the impact of new features used for the first time in this problem such as PageRank. A more detailed description of all these features is shown below:

- `[nterms]` Number of terms: Number of terms making up the query.
- `[Pop]` Number of Clicks Median: Median of the amount of clicks made during sessions associated with the query.
- `[DLev]` Levenshtein distance Median: Calculated between the query text and the text that represents the URL and title of the pages/sites selected in the associated query sessions.
- `[PageRank]` PageRank Median: each page/site selected has a PageRank coefficient. This feature represents the median value of this coefficient, considering all the pages/sites selected in the sessions where the query is formulated.
- ***n*CS**: Number of satisfied clicks: calculated as the quotient between the number of sessions where the query is formulated having less than n clicks over the total of query sessions. This feature was proposed for this problem by Liu et al. [20].
- ***n*RS**: Number of satisfied results: calculated as the quotient between the number of sessions where the query is formulated that only register clicks in the top-n results over the total of query sessions. This feature was proposed for this problem by Liu et al. [20].

4 Query intent data analysis

In this section we present a descriptive analysis of the features considered in our dataset meaningful for feature selection.

With respect to the features ***n***CS and ***n***RS proposed by Liu *et al.* [20] we will determine, based on an Information Gain criteria, for which values of ***n*** these variables have a better discriminatory behavior. Liu *et al.* used 2CS and 5RS in their work, showing a good discrimination power between the **navigational** and informational categories. In this paper, we also add the **transactional** category to the analysis.

As is depicted in Figure 3, a great proportion of the clicks is concentrated in the top-10 positions of the ranking, which is biased by the way the search engines return the results. Based on this observation, we set the search space for ***n*** to $\{1, 10\}$. Following, in Table 5 shows the Information Gain measure between each ***n-feature*** and the target variable.

n	1	2	3	4	5	6	7	8	9	10
***n*CS**	**0.57**	0.53	0.42	0.34	0.28	0.24	0.19	0.15	0.11	0.08
***n*RS**	0.71	**0.73**	0.67	0.63	0.61	0.58	0.57	0.52	0.48	0.43

Table 5: Information gain achieved for a number of values of ***n***CS y ***n***RS

Table 5 shows that `1CS` and `2RS` are the features with highest Information Gain values achieved. This reflects that the session data considered in our dataset has an important

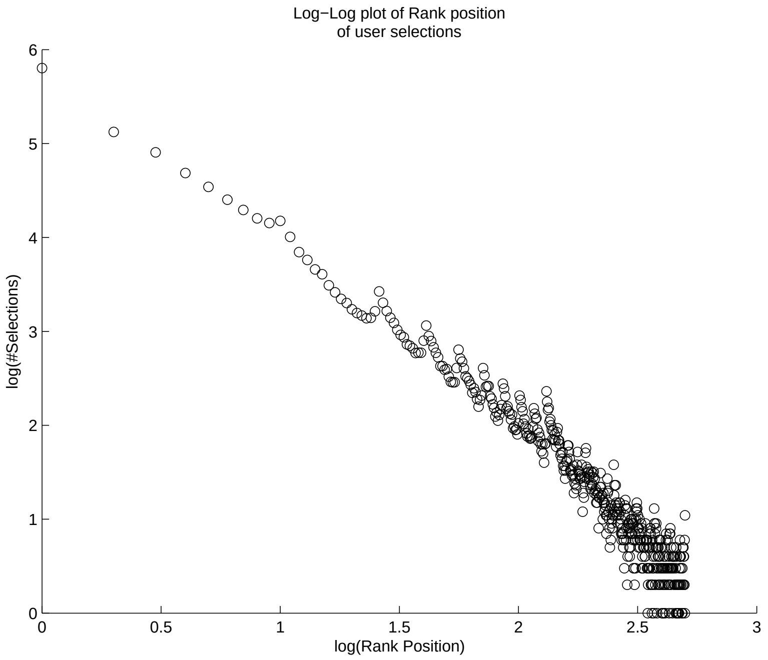
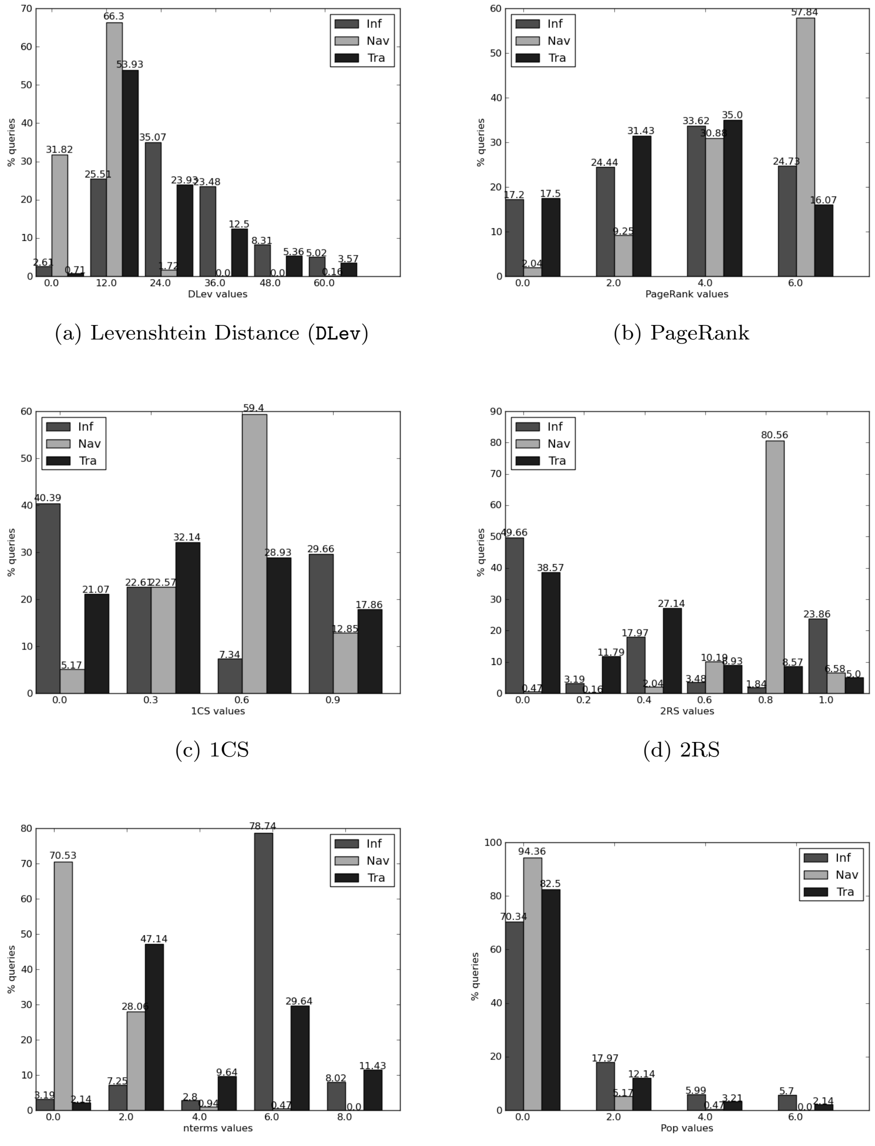


Fig. 3: Number of selections (clicks) for each rank position.

proportion of clicks concentrated in the top of the result lists. In the remainder of this paper, we only consider the features 1CS and 2RS for the analysis.

Figure 4 shows the distribution of values for each feature proposed in section 3.2 into the three types of query intent. Figure 4-(a) shows that the **navigational** queries obtain lower Levenshtein distance values than the other goals. The first two bins of the distribution concentrate over 90% of the queries with **navigational** intent, with informational and **transactional** capturing about 28% and 54% of the intent, respectively. On the other hand, Figure 4-(b) shows that the **navigational** queries obtain the highest values in PageRank. The last bin of the distribution concentrates 59.2% of the **navigational** queries, while the informational and **transactional** queries achieved 24.73% and 16.07% respectively. These first two figures indicate that the features PageRank and Levenshtein distance are very relevant for distinguishing **navigational** intent. This indicates to us is that the users that formulate **navigational** queries select the pages/sites whose descriptive text is very close to the descriptive text of the queries. Also, these pages/sites are usually very authoritative, reaching high PageRank values. Figure 4-(c) shows that the feature 1CS is relevant for informational and **navigational** intent. The first bin of the distribution concentrates close to 40% of the queries with informational intent, almost 20% more than those with **transactional** intent. The third bin concentrated close to 60% of the **navigational** intent, over 30% over the **transactional** intent. This indicates that at least 60% of sessions associated to almost 60% of **navigational** queries register just one click.

Figure 4-(d) shows that the 2RS feature is even more discriminating for **navigational** queries. The bin representing 2RS=0.8 concentrates over 80% of the **navigational** queries,



(a) Levenshtein Distance (`DLev`)

(b) PageRank

(c) 1CS

(d) 2RS

(e) Number of Query Terms (`nterms`)

(f) Median Click (`Pop`)

Fig. 4: Distribution of feature values grouping by user goal.

where only 8.75% are **transactional** and 1.84% are informational. The behavior shown by these features reflects that the users frequently select one page/site after submitting the query and they do it from the top two results of the list. Figure 4-(e) shows the distributions obtained for the number of query terms feature. The distribution shows that over 70% of the **navigational** queries have only one term (for example, queries where the term corresponds to the URL that the user wants to reach such as “amazon” or others), while in the case of **transactional** intent almost 50% of them have two or three terms (for example, *cue expressions* such as “download MP3” or others). Concerning queries with informational intent, over 80% of those are composed of six terms at least. Finally, Figure 4-(f) shows that the average number of clicks per session is not a very discriminating feature. The figure shows similar distributions, independently from the type of intent analyzed.

Table 6 shows the correlation coefficient values for each pair of variables considered in our dataset. The strongest correlation is found for the Levenshtein distance-number of query terms pair (0.7289); is obvious, since the more terms present in the query, the greater the Levenshtein distance between these terms and the descriptive texts of the pages/sites selected. Something similar occurs concerning the connection between the 2RS features and the number of terms in the query. Here, the correlation is negative (-0.5158) which indicates that when the queries have more terms, their associated sessions register clicks in places further down the list of responses.

	2RS	DLev	nterms	PageRank	Pop
1CS	0.4977	-0.2917	-0.2741	0.0205	-0.5025
2RS	-	-0.4057	-0.5158	0.2527	-0.2997
DLev	-	-	**0.7289**	-0.2425	0.1702
nterms	-	-	-	-0.3875	0.1196
PageRank	-	-	-	-	0.0284

Table 6: *Pearson* correlation factors for the features considered in our dataset.

5 Query intent detection methods

5.1 C 4.5-based method

A first approach we take is the use of a standard statistical learning algorithm as a C 4.5. We explore the use of decision trees for this problem in an earlier work [21]. The classifier can distinguish between informational, **navigational** and **transactional** intent, performing with noticeable precision. This model allows illustrating the usefulness of each feature studied and also obtaining an interpretable rule-based model by deriving conclusions from the generated tree connections. These connections show the capability of a set of features to discriminate between different query intent. Figure 5 depicts the best tree obtained, using the C 4.5 algorithm [24] available in WEKA [29], using 3-fold cross validation to improve the extent of results.

Considering the feature “nterms” at the root of the tree in Figure 5, it appears that the number of terms that compose a query is a very relevant feature. So, when the number of terms is one and the Levenshtein distance is less than 19.34 the intent is categorized as **navigational**. If it is higher than 24.85, it is informational. When the number of query terms is higher than four, the intent is categorized as informational. In the intermediate stretches of this variable, it is necessary to use other features to determine the user intent. In this sense, the features 1CS and 2RS are very relevant when the query has two or three terms. PageRank makes splits in the closest nodes to the leaves, which indicates to us that it is a very useful feature once the rest of the features have been used. We can also observe that the

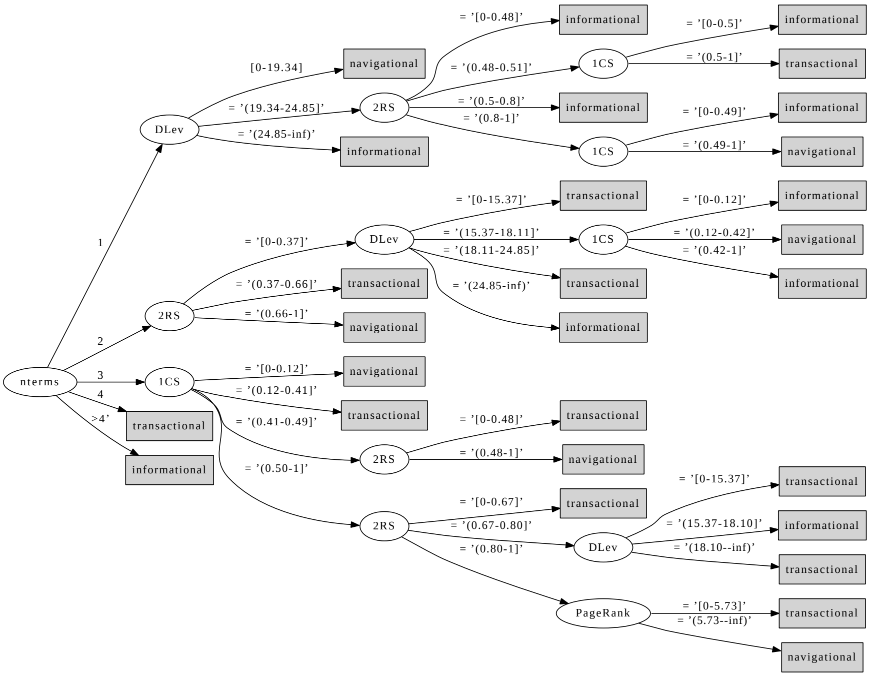


Fig. 5: Decision tree obtained using C 4.5

feature Median of Clicks was useless for the splits of the tree, which is why it is not used in the model obtained.

The simplest classification rules that can be induced from the tree consist of the conjunction of the variables `nterms` → `DLev` (**navigational** or informational) and `nterms` → `1CS` (**navigational** or **transactional**). In particular, the number of query terms feature is sufficient to determine a **transactional** or informational intent when the value four is taken, or values greater than four, respectively. The more complex rules concern the path made up by `nterms` → `1CS` → `2RS` → {`DLev` / `PageRank`} , which determine the depth of the tree.

5.2 Text-based methods

We introduce query intent classifiers corresponding to extensions of the vector space model. In this work, these query vector models are nourished by information about user selections, time spent on them and also the text from selected documents in the query log. We have decided to exploit the `Tf-Idf` vector model by adding information about user behavior. The vector space behind the query representations is determined by the terms appearing in queries and selected documents (content and title). The motivation for this approach is the observation that the content of the pages/sites attracts the user and leads him to invest more time in reading/using them. Is in this sense that this model seeks to increase the weight of terms with higher concentration of user's preferences. These methods were previously studied in our earlier work [22].

Tf-Idf Model

The `Tf-Idf` model, used frequently in information retrieval tasks such as page/site ranking for a given query, is also suitable for text categorization [14]. In this sense, the state-of-the-art shows us that a first naive approximation for building a vector representation of query intent could be based on the `Tf-Idf` model [27]. To this effect, we will use this model as *baseline*, building each query vector over a vector space made up by the terms appearing in past queries.

Given a query q (making a misuse of notation for q, equally referring for the user query and the query vector $\overrightarrow{q}$), let $q[i]$ be the weight of the component associated with the i-th term. Let $Tf_{i,q}$ be the number of occurrences that the i-th term has in q and let n_i be the number of queries in which the i-th term occurs. The weight $q[i]$ is given by the following expression:

$$q[i] = \frac{Tf_{i,q}}{\mathtt{MaxTf}_{\mathtt{l,q}}} \cdot \log(\frac{N}{n_i}), \tag{1}$$

where n_i denotes the number of queries in which the i-th appears, N represents the number of queries in the collection and $\mathtt{MaxTf}_{\mathtt{l,q}}$ is the frequency of the most frequent term in q.

The `Tf-Idf` model suggests that the terms that best describe a query are those with high discriminating capabilities. This capability is modeled according to the frequency of the term in the query and inversely with the frequency of the term in the query space. One of the *limitations* of this model is that the queries are usually made up by few terms, and therefore the vector that we can construct is often sparse.

We now discuss three variations of the `Tf-Idf` model that can be useful for constructing less scattered vector representations.

Tf-Pop Model

Baeza-Yates *et al.* [2] proposed a variation of the `Tf-Idf` model that considers the terms of the descriptive text of documents selected in query sessions. To do this, they tested the usefulness of the `Tf-Pop` representation, which replaces the `Idf` factor by the number of clicks that the document registers in the sessions associated to a given query. With this, they sought to give greater weight in the vector representation of a query to the terms that were used in documents that reflect a greater proportion of users preferences.

Given a query q, the weight $q[i]$ of the i-th term will be proportional to the number of occurrences of the term in each document selected in the query sessions of q (`Tf` factor) and will also be proportional to the fraction of clicks on each document u ($\mathtt{Pop}_{\mathtt{u,q}}$ factor). The factor $\mathtt{Pop}_{\mathtt{u,q}}$ is calculated from the quotient between the number of clicks concentrated in url u in sessions where query q is formulated and the total number of clicks registered in the sessions associated to q. Then $q[i]$ is given by the following expression:

$$q[i] = \sum_{\forall\, \mathtt{URL\, u}} \mathtt{Pop}_{\mathtt{u,q}} \cdot \frac{\mathtt{Tf}_{\mathtt{i,u}}}{\mathtt{Max}_{\mathtt{l}}\mathtt{Tf}_{\mathtt{l,u}}}. \tag{2}$$

According to this representation, a term will have a high weight for q when it has a high number of occurrences in the document (Tf factor) and the document registers more preferences in the query sessions of q ($\mathtt{Pop}_{\mathtt{u,q}}$ factor). Lets also observe that the component

$q[i]$ is calculated considering all the documents selected in query sessions associated to q, where the term is used.

Tf-Idf-Time **Model**

Claypool *et al.* [6] shows that a variable that correlates positively to the user preferences is the reading/visiting time or time-spent for a selected page/site. Claypool shows this through an experiment that measures the time spent in each page/site selected in the query sessions. Then, they compared these times with judgments made by experts on a collection of pages/sites considered in the study. They observed that there is a positive correlation between judgments and times. This observation allows suggesting that when the content of a page/site captures the attention of the user, the time spent there will increase.

With motivations based on Claypool's work, we will introduce a new query vector representation that considers the time spent on visiting the pages/sites selected in query sessions. The main idea consists of giving a higher weight to the i-th term in the query q ($q[i]$) as the term is more frequent and appears in more visited documents (documents with longer visit times). We will also consider the `Idf` factor, the same as in the first representation, assigning a high weight to the terms used in fewer documents and therefore have a high descriptive capacity.

Given a query q, the reading time will be denoted as t_u in the document u calculated for all the sessions where q is formulated (the sum of the times used to review u, considering all the qury sessions in which q is formulated). We will use t_S to denote the total duration of all the query sessions of q. Then, the component $q[i]$ is calculated by the following expression:

$$q[i] = \sum_{\forall\,\mathtt{URL}\,\mathtt{u}} \frac{\mathtt{Tf}_{\mathtt{i,u}}}{\mathtt{Max}_{\mathtt{l}}\,\mathtt{Tf}_{\mathtt{l,u}}} \cdot \frac{t_u}{t_S} \cdot \log(\frac{N}{n_i}). \tag{3}$$

Finally, the weight $q[i]$ is calculated considering all the documents selected in the query sessions of q where the term was used. The vector representation defined from Equation 3 will be denoted by $\mathtt{Tf} - \mathtt{Idf} - \mathtt{Time}$.

Tf-Idf-Pop-Time **Model**

A last extension of the `Tf-Idf` model consists of combining all the previously introduced factors in one single vector representation. The combination is done calculating each weight $q[i]$ through the product of the four factors discussed in this section, as the following expression shows:

$$q[i] = \sum_{\forall\,\mathtt{URL}\,\mathtt{u}} \frac{\mathtt{Tf}_{\mathtt{i,u}}}{\mathtt{Max}_{\mathtt{l}}\,\mathtt{Tf}_{\mathtt{l,u}}} \cdot \frac{t_u}{t_S} \cdot \log(\frac{N}{n_i}) \cdot \mathtt{Pop}_{\mathtt{u,q}}. \tag{4}$$

The weight $q[i]$ is calculated over the same collection of documents and queries used in the previous model. We will denote the vector representation defined from Equation 4 by $\mathtt{Tf} - \mathtt{Idf} - \mathtt{Pop} - \mathtt{Time}$.

Some Concerns About Query Vector Models

The fact that the amount of terms and the size of the log are connected by Heap's Law implies that a bigger log leads to a growing amount of terms. This behavior of the text, directly impacts over the dimensionality of the vector space of terms and, as a consequence, has an impact over the dimensionality of the query vectors. It will always be an important objective for the logs to be big enough to represent, in a more consistently way, the user's preferences. Due to this fact, it is advisable that the techniques used for building classifiers based on these vector representations be scalable in their dimensionality.

In this context, support vector machines (SVMs) have shown to be useful for processing high dimensional vectors, even when these vectors are sparse as in the case of text. Two works are especially important for the methodology adopted here: the first is one by Joachims [13] in which he showed in a theoretical way, that SVMs handle effectively large feature spaces in text classification. The second work, by Leopold & Kindermann [18] suggests that kernel selection is not as important as term-frequency transformations made in the text representation. Also, they studied several document representations using SVMs denoting an efficient performance without pre-filtering of terms (stopwords or lemmatization). Both works and others as well, position SVMs in a good level of efficiency and performance as compared to other techniques in the task of text classification, reasons why we will use them in our experiments.

5.3 Statistical learning strategies

We will use two learning strategies to deal with our dataset. A first strategy will consider that in the validation step each error weigths the same. A second strategy will consider cost sensitive learning. Due to class imbalance, our preliminar results [22] illustrate that the detection of **transactional** intents is particularly difficult. To tackle this problem we introduce a cost matrix that encodes the penalty for classifying **transactional** instances as **navigational** or informational intents. We set these penalty factors by considering that these kind of errors weight the twice an error between informational and **navigational** intents. We considered the same penalty factor for false positives and false negatives.

By declaring our cost matrix, the SVM strategy takes into consideration the penalty factors during model building, generating a model that has the lowest cost according to the cost matrix.

5.4 Combining different classifiers

To get better results in query intent detection, we explore the combination of different classifiers into a unified frame. This approach known as ensemble methods, has offered several advantages in different domains. We expect that the combination of our classifiers will offer precision improvements in our task due to the fact that our classifiers tend to be biased to specific query intents. This fact can be evidenced when we consider that the best features used for different query intents differ, as was shown in Table 4 and can be also observed in our C 4.5 tree in Figure 5. Thus, intuitively, the aggregation of the predictions of our multiple specific classifiers can produce improvements in classification accuracy.

Ensemble classifiers perform better than a single classifier when the base classifiers are independent of each other and the base classifiers outperforms random classification. Our earlier work indicates that our base classifiers achieve significant improvements over random classification (see [21, 22]). The first condition (independence) in general is much harder to prove but the state of the art indicates that improvements in classification accuracies can be observed even though the base classifiers are slightly correlated.

We will consider the combination of our text-based methods, and also an ensemble that combines text-based methods and the C 4.5 method. The impact of cost sensitive learning and the combination of cost sensitive and unsensitive learners will be also explored. The combination of our base classifiers will be conducted by aggregating their outputs in a very simple way. By calculating in a training query set the precision of each base classifier, we will weight each classifier outcome with the precision of each classifier. Then, the predicted label (the query intent) for a new testing query x will be decided by taking a majority vote, as follows:

$$C^*(x) = \mathtt{Vote}\{\mathtt{P}_{\mathtt{C_1}} \cdot \mathtt{C_1(x)}, \ldots, \mathtt{P}_{\mathtt{C_i}} \cdot \mathtt{C_i(x)}, \ldots, \mathtt{P}_{\mathtt{C_N}} \cdot \mathtt{C_N(x)}\},$$

where $\mathtt{C_i(x)}$ is the outcome of the $\mathtt{C_i}$ classifier (the predicted intent), $\mathtt{P}_{\mathtt{C_i}}$ is the precision of the i-*th* classifier for its specific outcome $\mathtt{C_i(x)}$ (the precision in the training set), and $C^*(x)$ is the outcome of the ensemble.

6 Experimental results

In this section we evaluate the performance of the classifiers proposed in Section 5 using the corresponding testing set. We calculate performance measures from a confusion matrix for each classifier, using 5-fold cross validation. Following a *one-against-all* approach, a predicted class was obtained for each test query evaluated, which was compared against the nominal class and finally, the result was added to a confusion matrix. After the complete evaluation of all test queries and using the confusion matrix generated, we calculate precision, FP rate, recall and F1 performance measures.

Additionally, to illustrate the tradeoff between the predictive and discriminative ability of the classifiers, we will use ROC curves (**R**eceiver **O**perating **C**haracteristics), allowing us to reach conclusions about the performance of the proposed models. We have selected a visualization technique such as ROC instead of other evaluation techniques, such as precision-recall curves, mainly because it has shown good results in classification problems, even when there are class skews as Fawcett [10] indicates.

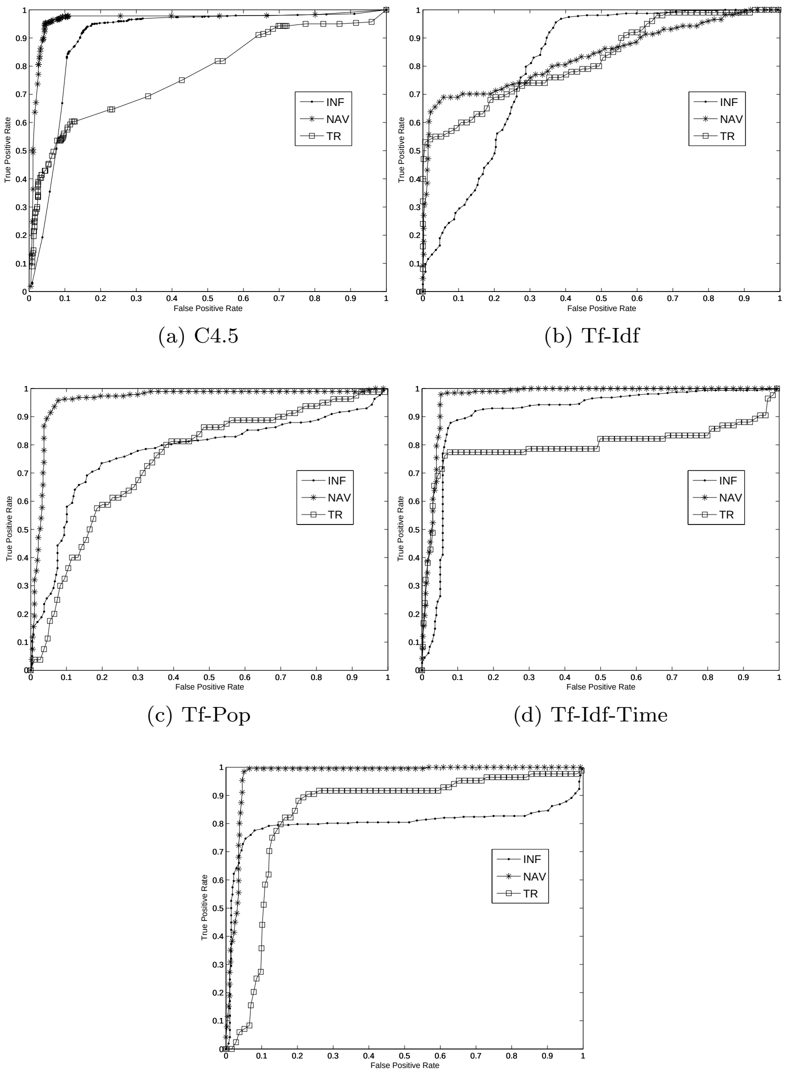
ROC curves are built by using the *TP Rate* measure plotted on the Y axis and the *FP Rate* measure plotted on the X axis. In this way, a graphic comparison between benefits (Y axis) and costs (X axis) is allowed. From each of the curves plotted in this experiment, the AUC measure (the area below the curve) was calculated. This measure estimates the probability that the classifier can correctly classify a positive instance higher than a negative one. When the AUC value increases, its average performance also increases. Table 7 shows the results of this evaluation. The ROC curves are depicted in Figure 6.

Table 7 shows us that the studied classifiers achieve good results, especially for informational and **navigational** intent. Notice that bold fonts indicate the best result in F_1 measure and AUC for each category. In the case of **transactional** goals, the five classifiers evaluated obtained lower performances. The results indicated by the label *Weighted* were calculated by the weighted average of the three base classifiers regarding the fraction of queries of each intent (52%, 33%, and 15% for the informational, **navigational** and **transactional** intents, respectively).

Given the **informational** category and considering the F_1 and AUC measures, the classifiers that achieved the highest values are `C 4.5`, obtaining 90% and 89.7% in F_1 measure and AUC measure respectively, and `Tf-Idf-Time`, attaining a 89% and 91% in F_1 measure

		Evaluation Measures				
		Recall	FP Rate	Precision	F_1	AUC
Inf.	`C 4.5`	0.936	0.163	0.866	**0.901**	0.897
	`tf-idf`	0.945	0.364	0.746	0.835	0.807
	`tf-pop`	0.711	0.184	0.812	0.758	0.773
	`tf-idf-time`	0.907	0.138	0.881	0.894	**0.912**
	`tf-idf-pop-time`	0.756	0.061	0.932	0.835	0.801
Nav.	`C 4.5`	0.956	0.052	0.901	0.927	0.954
	`tf-idf`	0.666	0.041	0.872	0.755	0.837
	`tf-pop`	0.791	0.037	0.913	0.848	0.960
	`tf-idf-time`	0.984	0.065	0.878	0.928	**0.972**
	`tf-idf-pop-time`	0.994	0.053	0.901	**0.945**	**0.972**
Tr.	`C 4.5`	0.393	0.027	0.705	0.505	0.763
	`tf-idf`	0.530	0.004	0.963	**0.683**	0.825
	`tf-pop`	0.437	0.141	0.339	0.383	0.744
	`tf-idf-time`	0.392	0.017	0.785	0.523	0.801
	`tf-idf-pop-time`	0.690	0.121	0.487	0.571	**0.836**
Weig.	`C 4.5`	**0.861**	0.105	0.853	0.849	0.895
	`tf-idf`	0.790	0.203	0.820	0.785	0.819
	`tf-pop`	0.696	0.129	0.774	0.731	0.830
	`tf-idf-time`	0.855	0.095	**0.865**	**0.849**	**0.915**
	`tf-idf-pop-time`	0.824	**0.067**	0.855	0.831	0.862

Table 7: Performance evaluation of the proposed base classifiers.



(a) C4.5

(b) Tf-Idf

(c) Tf-Pop

(d) Tf-Idf-Time

(e) Tf-Idf-Pop-Time

Fig. 6: ROC curves for the base classifiers.

and AUC respectively. Regarding the classification tree built by `C 4.5` and shown in figure 5, these experimental results suggest that nterms, by itself, is a useful feature for identifying this kind of intent, above all when the query is composed by over 4 terms. Regarding this model, and in union with nterms, another particularly relevant feature was Dlev, which allowed showing that longer distances between document title and query text, characterize **informational** intent. The observation presented above suggests that more general queries in comparison with more straight ones (as seen with **navigational** goals), present a greater diversity in the selected documents, which in turn allows us to substantiate one of the main characteristics of this kind of intent. On the other hand, regarding the vector model mentioned above, an important aspect to be under consideration is that when the popularity factor is added to the representation (namely `Tf-Idf-Pop-Time`), the performance falls off, showing in this sense, that this feature does not seem to be quite useful for this task. Besides, when making a comparison between the performances of the F_1 and AUC measures for the model with and without the time factor (`Tf-idf` and `Tf-Idf-Time` respectively), a noticeable decrease is noticed. Such behavior may be notoriously observed in the AUC measure, where the performance falls off about 11 percentual points. Finally, considering this kind of goals, the representation of text using the coefficients Tf , Idf, nterms and DLev, results to be of fundamental importance, allowing confirming previously made assumptions concerning the close relationship between text information and **informational** goals.

Considering **navigational** intent, `Tf-Idf-Pop-Time` is the classifier that obtained superior results in both AUC and F_1 measures. Besides, considering both measures, the nearest model to the previously mentioned was `Tf-Idf-Time`, which presented nearby results. It is important to notice that clicks to be a useful source of information and the same observation applies to the time, which not by chance appears in both representations, denoting an interesting level of influence in the discrimination of this type of goals. As well as the mentioned models, `C 4.5` achieved similar performance results. The tree obtained and depicted in Figure 5, shows that short queries (low amount of terms) having a close textual relationship (low Levenshtein distance), point to **navigational** intent. Another relevant characteristic, present in the tree from the figure, is that it classifies short queries, having a greater amount of associated sessions with selections on top two results as **navigational** queries (queries with higher values for `2RS`). Essentially, the `C 4.5` model agrees with the assumption about the lower dispersion in user selections and higher position of selections under **navigational** intent, denoting in this way a more accurate objective in queries for this kind of need.

Table 7 shows that the five classifiers decreased their performance when attempting to detect **transactional** goals. Even when the global performance falls off, it is an unfair situation to base the analysis only in measures such as precision, recall or F_1, because there exists a notorious unbalance in the classes, particularly in the **transactional** class. Due to this fact, the performance measure considered for this class will be the area under the ROC curve (AUC). The `C 4.5` tree depicted in Figure 5, shows that the query length (nterms) does not seem to be a very useful feature to discriminate between **informational** and **transactional** classes. For example, the root of the tree presents the rules [`nterms` $>$ 4 → ***informational***] and [`nterms` $=$ 4 → ***transactional***], which seem to be quite similar and make it very difficult to establish further conclusions. Also, it is a difficult task to discriminate queries based on the distance between the query and the selected documents, mainly because the Levenshtein distances presented by **informational** and **transactional** queries are comparable, as is shown in the branch [`nterms`→`2RS`→ `DLev`] on Figure 5. Additionally, the histogram plot, depicted in Figure 4a, shows a especially similar behavior of the DLev feature in **transactional** and **informational** categories when the distance rises from 24 to higher values. Regarding vector models, the highest performances were achieved by Tf-Idf-Pop-Time in first place and following it, the baseline model. Considering the baseline model and incorporating popularity and reading time information (Tf-Idf-Pop-Time), the AUC measure increases from 82.5% to 83.6%. However, replacing the Idf factor by the Pop factor, causes a decrease in the

AUC measure to 74%. This observation suggests that in effect, the text (Tf and Idf factors), the reading time and the popularity are useful and representative features of **transactional** needs.

As is shown in Figure 6, the **transactional** intent was the most difficult to detect, thus suggesting that some other information source might be missing. Even considering this, it is important to contemplate that the most feasible performance measure in this work is the AUC, because precision and recall are sensitive to class skew. For this intent, the best performance measures are at least one percentual point below the results achieved in the past two goals and also, both `C 4.5` and `Tf-Pop` obtaining the lowest AUC value. These results indicate that some **transactional** features are missing and also that the popularity factor is not as useful as initially expected. Additionally, the baseline obtained a comparable performance to `Tf-Idf-Pop-Time`. This is an interesting result, since it allows suggesting that query terms are useful for the detection of this user intent.

For the **informational** goal, the `C 4.5` and the `Tf-Idf-Time` models are more useful in predicting the given intent. This indicates first, that text from queries (`nterms`) and document title (`DLev`) are the most descriptive features for this goal and second, that the text from documents and the reading-time together, are very powerful for predicting this kind of intent. In the case of the **navigational** goal, all the models shown obtained their best performance in this task. Considering the `C 4.5` model, the results suggest that `nterms`, `DLev`, clicks (`Pop`) and position (`2RS`) of search results are quite relevant for this kind of identification. Now, considering the vector models, an interesting result is that the lower performance was achieved by the baseline, indicating that the use of text only is not sufficient to identify the **navigational** intent.

To address the limited performance of our base classifiers for **transactional** intent detection we tested the cost-sensitive learning strategy discussed in Section 5. Our cost matrix includes penalties for errors involving the **transactional** intent, trying to get better results in this intent that in fact is the one which achieved the lower performance using our base classifiers, as was shown in Table 7. These results are showed in Table 8.

Table 8 shows that the use of cost sensitive learning for the **transactional** intent detection allows to improve the performance in this specific class but diminishing the performance in the other classes. Some of the improvements obtained in the **transactional** intent are very significant. For example, the performance achieved by the Tf-Idf-Pop-Time classifier outperforms from 10 to 15 percentual points its original no sensitive version. On the other hand, the performance achieved by detecting **informational** and **transactional** intents tend to decrease from 2 to 8 percentual precision and recall points. The balance between the classes can be observed in the macro averaged results, indicated by the label *Weighted*. Bold fonts indicates the best performance results achieved in each measure. At the macro average level, the five best results are achieved by the text-based methods tf-idf-time and tf-idf-pop-time. Tf-idf shows a good discriminative performance for the **transactional** intent, achieving an almost perfect false positive rate and the best precision rate for this intent but with low recall. On the other hand, the best recall performance is achieved by tf-idf-pop-time, fact that affects the performance precision.

In spite of the previous fact, the best precision-recall balance is achieved by tf-idf-pop-time, which reaches the high F-measure value in the **transactional** intent. A similar performance

		Evaluation Measures				
		Recall	FP Rate	Precision	F_1	AUC
Inf.	`C 4.5`	0.912	0.171	0.861	0.885	0.866
	`tf-idf`	**0.930**	0.274	0.740	0.824	0.804
	`tf-pop`	0.702	0.193	0.803	0.749	0.769
	`tf-idf-time`	0.904	0.146	0.872	**0.887**	**0.903**
	`tf-idf-pop-time`	0.742	**0.073**	**0.920**	0.821	0.796
Nav.	`C 4.5`	0.944	0.059	0.879	0.910	0.950
	`tf-idf`	0.645	0.043	0.851	0.733	0.834
	`tf-pop`	0.785	**0.041**	**0.902**	0.839	0.952
	`tf-idf-time`	**0.981**	0.068	0.871	0.922	0.953
	`tf-idf-pop-time`	0.978	0.059	0.893	**0.933**	**0.961**
Tr.	`C 4.5`	0.442	0.017	0.700	0.541	0.781
	`tf-idf`	0.567	**0.003**	**0.925**	0.703	0.833
	`tf-pop`	0.593	0.121	0.583	0.587	0.760
	`tf-idf-time`	0.490	0.013	0.744	0.590	0.813
	`tf-idf-pop-time`	**0.782**	0.098	0.648	**0.708**	**0.868**
Weig.	`C 4.5`	0.852	0.110	0.842	0.842	0.880
	`tf-idf`	0.781	0.157	0.804	0.776	0.818
	`tf-pop`	0.713	0.132	0.802	0.754	0.828
	`tf-idf-time`	**0.867**	0.100	0.852	**0.854**	**0.906**
	`tf-idf-pop-time`	0.825	**0.072**	**0.870**	0.841	0.861

Table 8: Performance evaluation of the classifiers using cost sensitive learning.

behavior can be observed in the **navigational** intent detection, where tf-idf-pop-time achieves also the best precision recall balance. However, for the detection of **informational** intents the best precision-recall balance is achieved by tf-idf-time, illustrating that the use of clicks as a source of user feedback introduces noise in this classifier. However, **navigational** and **transactional** intents are better detected by introducing time as a source of user feedback suggesting that the learning process can leads to the creation of classifiers that are focused on specific intents. In order to take advantage of this fact, we explore a mixture of experts strategy which allow the creation of classifier ensembles.

Our ensemble strategy picks the best base classifier for each intent. We use a combination rule which is based on the weighted majority voting, where the weight is assigned to the classifier according to a performance measure. As was discussed in Section 5, we will weight each vote by considering the precision reached for the base classifier in the predicted intent. As in the previous experiments, we use five fold cross validation but with an important modification. We considered two testing phases, a first one that allow the evaluation of each base classifier for its inclusion in the ensemble. A second one, independent of the previous evaluation, was conducted over the ensemble, considering query buckets which were not included in the training/validation/testing process involved in the creation of the base classifiers. The data partition strategy is described in Table 9.

Partition	Training	Validation	Testing	Ensemble Testing
(1)	$\{S_1, S_2\}$	S_3	S_4	S_5
(2)	$\{S_2, S_3\}$	S_4	S_5	S_1
(3)	$\{S_3, S_4\}$	S_5	S_1	S_2
(4)	$\{S_4, S_5\}$	S_1	S_2	S_3
(5)	$\{S_5, S_1\}$	S_2	S_3	S_4

Table 9: Data partition strategy using five folds.

As Table 9 suggests, a new learning process was conducted for each fold. This was done by considering cost sensitive learning and cost no sensitive learning. For each strategy, we pick the best classifier per intent, according to the average precision reached across the five data partitions. We summarize these results in Table 10.

			Evaluation Measures			
Strategy	Intent	Best classifier	P@Inf	P@Nav	P@Tr	Weig.
Basic	Inf.	`tf-idf-pop-time`	**0.859**	0.852	0.411	0.789
	Nav.	`tf-pop`	0.798	**0.912**	0.282	0.758
	Tr.	`tf-idf`	0.661	0.844	**0.918**	0.759
Cost sensitive	Inf.	`tf-idf-pop-time`	**0.838**	0.829	0.580	0.796
	Nav.	`tf-pop`	0.723	**0.839**	0.519	0.730
	Tr.	`tf-idf`	0.657	0.825	**0.930**	0.753

Table 10: Best classifiers selected for the creation of the ensembles.

As Table10 shows, the performance of the base classifiers decreases from 3 to 8 precision percentual points. However, these results show consistency with the results showed in Tables 7 and 8 in the following senses. First, the best classifiers for each intent are the same, illustrating the fact that the use of two data buckets for training instead of three training data buckets affects the performance without changing the performance relative order between the classifiers. Second, the use of cost sensitive learning allows to obtain improvements for transactional intent detection affecting the performance in the detection of informational and navigational intents. In this case, a significant precision decrease around 8 percentual points is observed for navigational intents, being less important this decrement for informational intent detection.

Now we evaluate the performance of the combination of these classifiers, by creating two ensembles, one based on the basic classifiers and another one based on the cost sensitive learners. The majority vote function for each of these ensembles considers the weights showed in Table 10 to decide the outcome for each testing query. The evaluation of these ensembles is conducted by using the ensemble testing queries, for each partition of the five fold cross validation strategy considered in this experiment, according to the data partition schema showed in Table 9. Then, the performance measures are calculated across the five data partitions, as was also done in the experiments showed in Tables 7 and 8. The results for these evaluations are shown in Table 11.

		Evaluation Measures				
		Recall	FP Rate	Precision	F_1	AUC
Inf.	`Baseline`	0.758	0.088	0.932	0.836	0.811
	`Basic`	0.812	0.037	0.945	0.873	0.863
	`Cost sensitive`	0.778	0.069	0.928	0.846	0.836
Nav.	`Baseline`	0.788	0.072	0.895	0.838	0.807
	`Basic`	0.821	0.012	0.948	0.879	0.912
	`Cost sensitive`	0.793	0.035	0.918	0.850	0.865
Tr.	`Baseline`	0.613	0.099	0.614	0.613	0.648
	`Basic`	0.602	0.090	0.978	0.745	0.812
	`Cost sensitive`	0.664	0.080	0.967	0.787	0.840
Weig.	`Baseline`	0.746	0.084	0.872	0.804	0.785
	`Basic`	**0.783**	**0.036**	**0.950**	**0.859**	**0.871**
	`Cost sensitive`	0.765	0.059	0.930	0.840	0.846

Table 11: Performance evaluation of the ensemble classifiers.

Table 11 shows that the use of ensembles allows to reach significant performance improvements. We have included as a baseline the best single classifier, which corresponds to tf-idf-time, being this one the classifier that achieves the best performance results at macro average level. The ensembles achieve the best results in all the comparisons, being possible the identification of improvements in discriminative capacities, reducing false positive rates, and also in the intent detection task itself, increasing recall and precision by several percentual points. Thus, F measures achieve the best performance, being these results very close to the 90% of balance in the three intent detection cases. Some improvements are very relevant. For example, the cost sensitive ensemble outperforms the baseline by almost 20 F-measure percentual points for the transactional intent detection task. However, at the big picture level, the basic ensemble reaches the best results at macro average level, which are indicated in bold fonts. This fact can be explained from the distribution of informational, navigational and transactional testing queries, which emphasizes the performance in informational and navigational intents, task that is well done by the basic ensemble.

7 Conclusion

In this paper, we worked on the problem of automatically detecting query intent. Following the taxonomy of Web searches proposed by Broder [4], we have developed a new dataset with almost 2,000 queries tagged in informational, navigational and transactional categories. Using a real-world query log, we calculated features for each of these queries; we have as well determined optimal values for the parametric nCS and nRS features using an information theory approach. The dataset also considers features not previously studied in this problem, such as PageRank and the Levenshtein distance. The usefulness of hyperlinks and text to automatically identifying user intent is shown in this way. Besides the construction of features, models from two different worlds were built. First, a tree classifier showing some hierarchical relationships between features and second, vector representations, were constructed based on text and query log mining measures, thus extending the vector space model.

To determine the relevance of the text in the detection of user's goals, we evaluated the performance of four classifiers using vector representations of queries. Here, we used the `Tf-Idf` model over the query space as the baseline. Three extensions of the `Tf-Idf` model based on query log mining were studied in this work. To test the vector classifiers, we employed Support Vector Machines, allowing us to determine hyperplanes of maximum separability among the three classes analyzed.

The experiments shown in section 6 allowed us to compare the five classifiers discussed. The experimental results showed that the classifiers reached different performances depending on the type of intent detected. Concerning informational intent, the best results were obtained using the three new classifiers proposed in this paper, achieving an almost perfect performance in the case of C 4.5. The text was shown to be very useful in the case of navigational intents. The Tf-Idf-Time and Tf-Idf-Pop-Time classifiers attained practically optimum results (close to 95%).

When studying transactional intent, we confirmed that this type of intent is the most difficult to detect. The five classifiers evaluated lowered their performances, with the best achievement belonging to the baseline which utilized query terms. In this sense, we confirmed Jansen's results *et al.* [12], who faced the problem using the three categories. They also showed the usefulness of text from the queries to detect transactional intents. However, unlike Jansen's work, in this paper the proposed classifiers were constructed without human intervention of any kind. The incorporation of user's preferences to determine descriptive text from each query is a novel aspect that sets this paper apart from its predecessors.

Finally, we explore different learning strategies to address the problem impossed by the detection of transactional intents. In a first approach, we define a cost matrix which imposses a penalty factor to learning errors which involve transactional intents. This cost sensitive learning strategy allows to get better results for transactional intent detection but decreasing the performance in the other classes. To address this limitation, we explore the use of ensemble methods, created from a set of base classifiers previously explored in our earlier work. We created our ensembles by picking the best classifier for each query intent, performing a combination of them by using a majority vote function, that considers weights for each classifier outcome according to a performance measure calculated at the training step. Two ensembles were evaluated, a first one based on the unsensitive cost classifier and a second one based on cost sensitive classifiers. By comparing the performance of the ensembles with a baseline, which corresponds to the best single classifier at macro average level, we found that the use of ensembles offers several advantages, being possible the creation of a mixture of experts, where each base classifier emphasize specific variables or information sources to get better intent detection results.

Acknowledgements

We thank the anonymous reviewers for useful suggestions. Mr. Mendoza was supported by CONICYT project FONDECYT 11121435, Mr. Zamora was supported by MECESUP project FSM0707 and Mr. Allende was supported by CONICYT project FONDECYT 1110854.

References

1. Azin Ashkan, Charles Clarke, Eugene Agichtein, and Qi Guo. Classifying and characterizing query intent. In *Proceedings of the European Conference of Information Retrieval, ECIR, Toulouse, France*, pages 578–586. Springer, 2009.
2. Ricardo Baeza-Yates, Liliana Calderón-Benavides, and Cristina González-Caro. The intention behind web queries. In *Proceedings of the International Symposium on String Processing and Information Retrieval, SPIRE, Glasgow, Scotland, UK*, pages 98–109. Springer, 2006.
3. Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. Improving search engines by query clustering. *J. Am. Soc. Inf. Sci. Technol.*, 58(12):1793–1804, 2007.
4. Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.
5. Liliana Calderón-Benavides, Cristina González-Caro, and Ricardo Baeza-Yates. Towards a deeper understanding of the user's query intent. In *Workshop on Query representation and Understanding, SIGIR workshop, Geneva, Switzerland*, 2010.
6. Mark Claypool, David Brown, Phong Le, and Makoto Waseda. Inferring user interest. *IEEE Internet Computing*, 5(6):32–39, 2001.
7. Wilburt Daelemans, Jamie Zavrel, Kurt van der Sloot, and Alex van den Bosch. Timbl: Tilburg memory based learner, reference guide. Technical Report Technical Report Series 10-01, ILK Research Group, 2010.
8. Zhang Dou, Li Zhang, Xiao Yuan, and Feng Liu. Automatic user goals identification based on anchor text and click-through data. In *Proceedings of the Conference of Web Information System and Application, WISA, Xian, China*, 2008.
9. Wei Fan, Mark Gordon, and Park Pathak. Discovery of context-specific ranking functions for effective information retrieval using genetic programming. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):523–527, 2004.
10. Tom Fawcett. An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874, 2006.
11. Carlos Hurtado and Marcelo Mendoza. Automatic maintenance of web directories by using web browsing data. *J. Web Eng.*, 10(2):153–173, 2011.
12. Bernard Jansen, Danielle Booth, and Amanda Spink. Determining the informational, navigational, and transactional intent of web queries. *Inf. Process. Manage.*, 44(3):1251–1266, 2008.
13. Thorsten Joachims. A statistical learning model of text classification for support vector machines. In *Proceedings of the Conference on Research and Development in Information Retrieval, SIGIR, New Orleans, LA, USA*, pages 128–136. ACM Press, 2001.
14. Thorsten Joachims. *Learning to Classify Text using Support Vector Machines.* Kluwer, 2002.
15. In-Ho Kang. Transactional query identification in web search. In *Proceedings of the Asia Information Retrieval Symposium, AIRS, Jeju Island, Korea*, pages 221–232. Springer, 2005.
16. In-Ho Kang and Gil-Chang Kim. Query type classification for web document retrieval. In *Proceedings of the Conference on Research and development in information retrieval, SIGIR, Toronto, Canada*, pages 64–71. ACM, 2003.
17. Uichin Lee, Zhenyu Liu, and Junghoo Cho. Automatic identification of user goals in web search. In *Proceedings of the International Conference on World Wide Web, WWW, Chiba, Japan*, pages 391–400. ACM, 2005.
18. Edda Leopold and Jörg Kindermann. Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, 46(1-3):423–444, 2002.
19. Tie Yan Liu, Jian Xu Tie Qin, Wu Xiong, and Hu Li. Letor: Benchmark dataset for research on learning to rank for information retrieval, 2007.

20. Yiqun Liu, Min Zhang, Liyun Ru, and Shaoping Ma. Automatic query type identification based on click through information. In *Proceedings of the Asia Information Retrieval Symposium, AIRS, Singapore*, pages 593–600. Springer, 2006.
21. Marcelo Mendoza and Juan Zamora. Building decision trees to identify the intent of a user query. In *Proceedings of the International Conference on Knowledge-based and Intelligent Information and Engineering Systems, KES, Santiago, Chile*, pages 285–292. Springer, 2009.
22. Marcelo Mendoza and Juan Zamora. Identifying the intent of a user query using support vector machines. In *Proceedings of the International Symposium on String Processing and Information Retrieval, SPIRE, Saariselka, Finland*, pages 131–142. Springer, 2009.
23. Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *Proceedings of the International Conference on Scalable Information Systems, InfoScale, Hong Kong, China*, page 36. ACM Press, 2006.
24. John Ross Quinlan. *C4.5: programs for machine learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
25. Filip Radlinski, Martin Szummer, and Nick Craswell. Inferring query intent from reformulations and clicks. In *Proceedings of the International Conference on World Wide Web, WWW, Raleigh, North Carolina, USA*, pages 1171–1172, 2010.
26. Daniel Rose and Danny Levinson. Understanding user goals in web search. In *Proceedings of the International Conference on World Wide Web, WWW, Manhattan, NY, USA*, pages 13–19. ACM, 2004.
27. Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.
28. Ji-Rong Wen, Jian-Yun Nie, and Hong-Jiang Zhang. Clustering user queries of a search engine. In *Proceedings of the 10th international conference on World Wide Web, WWW, Hong Kong, China*, pages 162–168. ACM, 2001.
29. Ian Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann, 1st edition, 1999.

View publication stats