A Search Engine based on Query Logs, and Search Log Analysis at the University of Sunderland

Michael Oakes, Yan Xu

University of Sunderland, Dept. of Computing, Engineering and Technology, DGIC, St. Peter's Campus, Sunderland SR6 0DD, England
{michael.oakes,yan.xu-1}@sunderland.ac.uk

Abstract. This work describes a variation on the traditional Information Retrieval paradigm, where instead of text documents being indexed according to their content, they are indexed according to the search terms previous users have used in finding them. We determine the effectiveness of this approach by indexing a sample of query logs from the European Library, and describe its usefulness for multilingual searching. In our analysis of the search logs, we determine the language of the past queries automatically, and annotate the search logs accordingly. From this information, we derive matrices to show that a) users tend to persist with the same query language throughout a query session, and b) submit queries in the same language as the interface they have selected, except in a large number of cases where the English interface is used to submit Latin queries.

ACM Categories and Subject Descriptors: H Information Systems; H3 Information Storage and Retrieval; H3.3 Information Search and Retrieval; Search Process.

Free Keywords: Automatic Language Identification, Multilingual Search Engine, Query Logs.

1 Introduction

A number of authors have previously used search logs to improve search engine performance. Hoy and Lyu (2004) used search logs for relevance feedback, a process whereby judgements on the quality of documents initially retrieved by a search engine are used to retrieve even more relevant documents at a second stage. Normally these judgements would be made by the user who had submitted the original query, a task which many search engine users find laborious. In this approach however, the judgements are effectively made by previous users who, as recorded in the search logs, chose to either download or not download the same documents. Cui et al. (2003) identified correlations between query terms and document terms by analysing search logs with a probabilistic model.

The hypothesis behind search log-based approaches to search engine design is that previous users' choices are of interest to new users who input similar queries. We take an extreme position on this, by rather than indexing documents based on their content as in conventional search engines, we index documents solely with the terms past users have used in searching for them, as found in the search logs. Collated under each downloaded document ID will be the terms of every query ever submitted in a session leading up to the downloading of that document. Thus query terms from separate sessions leading to the retrieval of the same document will all become index terms for that document.

This approach is beneficial for multilingual searching, since each previously downloaded document is indexed by all search terms which have ever been used in a search for that image, irrespective of which language they were in. Thus a document might be indexed by search terms in various languages. The user can then submit search terms in the language of his or her choice. If some of the index terms of a relevant document are in that language, there is a chance that these will match the query terms, allowing the user to retrieve this document in its original language. The limitation of this query log approach is that if previous users have never downloaded a particular image, then that image can never be retrieved by this technique. This is a practical limitation of the training data, not a theoretical limitation, but shows the need for large amounts of search log training data.

2 Implementation of a Search Engine Based on Query Logs

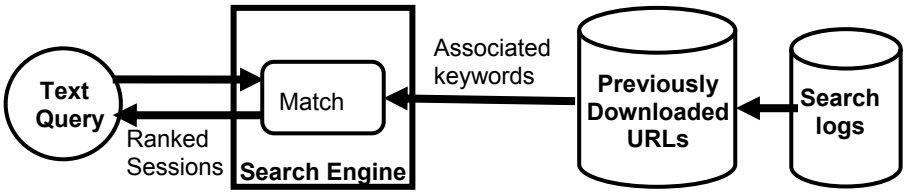
The first step in the analysis of query logs necessary to index search engine documents is to be able to identify the start and end of each query session. This is non-trivial in some data sets, but a unique ID is given to each

session in the LogCLEF search logs. Query records for which the seventh field was "search xxx", "available_at" or "see_online" were assumed to indicate the downloading of the relevant URL cited in field 12. These URLs would then be indexed by all the search terms submitted in either that session or any other session which resulted in the downloading of that same URL.
The queries were not stop listed, since they were multilingual, which would require a stop list for each language and increase the danger that a stop word in one language might be meaningful in another. Instead, a weighting scheme was used so that each query term would be given a weight reflecting its importance with respect to each document. This measure, called TF.IDF, assigns low weights to words with low information content. The formula is as follows:

$$w_{kd} = f_{kd} \cdot \log\left(\frac{NDoc}{D_k}\right)$$

In a conventional search engine, where a document is indexed based on its content, f_{kd} is the frequency of term k in document d, D_k is the number of documents which include that term, $NDoc$ is the total number of documents in the collection, and w_{kd} is the weight reflecting the importance of term k to document d. In our approach, f_{kd} is the number of times query term k has been used in a search for document d. D_k is number of documents which have been accessed using that query term, and $NDoc$ is the total number of documents downloaded at some stage in the search logs. Thus w_{kd} is the weight reflecting the typicality of term k with respect to session d. The highest TF.IDF scores are given to those terms which are often used in searches for the document of interest, but are not used in searching for many other documents.
Now that all the previously downloaded documents have been indexed, we can match the queries of future users against the document index terms using the cosine similarity coefficient, as in a conventional search engine. The documents are ranked according to their similarity to the user's query, and the best matching documents are presented to the user. The architecture of our query log-based search engine is shown in Figure 1.

Figure 1. Architecture of the Query Log-Based Search Engine



3 Evaluation of the Search Engine

For training the search engine we used 1399747 records (the first 75%) and for testing we used 466582 (the remaining 25%). It was necessary to sort the logs into chronological order in order to separate the test and the training set. If we define a search session as all records with a common session ID, the LogCLEF file contained 225376 sessions. Average session length according to this definition was 1399747 / 225376 = 6.21 lines. For the search engine approach, however, we defined a search session as being a sequence of records with the same ID which included one of the following commands, indicative of the retrieval of a relevant record: search xxx, see online, available at. Some of these sessions recorded more than one retrieved URL.
For each URL in the test set, we combined all the query terms used in the same session into a single text query. We then matched this query against each of the query term sets collated under URLs in the test set. We assumed that the URL retrieved in the test session was the gold standard, and wished to determine the search engine's ability to retrieve the session (if there was one) with the same URL from the training set. There were 8586 URLs in the test set sessions, and 284 of them matched previous records retrieving the same URL in the top 100 best matching training set sessions. Thus the percentage of matched URLs was 3.32%.

4 Analysis of the Search Logs

Since the indexing process described in section 2 requires the set of search logs to be read in, it would be possible to incorporate modules (at present written as separate programs) into the search engine for the analysis of these logs. The overall procedure we have followed for search log analysis is as follows:
The search logs are read in, and we find the most likely language of the query terms on each line. Each line of the search log is then annotated with the name of the query language on that line, or "null" if no query is present, as described in section 5. The frequencies of each query language used in the first 100000 lines of the logs are given in Section 6. Given the sequence of query languages in the logs, we determine the likelihood of a query in one language (or a new session) being followed by a query in each of the other languages, another query in the same language, or the end of the session. This program is described in section 7. For each interface language, we determine the frequency of the query languages used, as described in section 8.

5 Automatic Language Identification

Souter et al. (2004) describe a technique for automatic language identification, based on trigram (sequences of three adjacent characters) frequencies. They found that using knowledge of the frequencies of the trigrams occurring in 9 different languages, they were able to accurately identify a language from a sample of just 175 characters. We implemented this method to determine the language of each of the queries recorded in the search log. The method gave subjectively reasonable results, although we anticipate less than 100% accuracy, since our query samples were shorter than those used by Souter et al., typically about two words long.
To estimate the trigram frequencies typical of a set of languages, we used the Europarl corpus which contains transcripts of meetings of the European Parliament in each of 11 languages. The languages of Europarl are Danish, Dutch, English, Finnish, French, German, Greek, Italian, Portuguese, Spanish and Swedish.
We counted the frequency of each trigram in a sample of just over one million characters for each language. The frequency of each trigram was divided by the total number of characters (minus two) in the sample of that language, to give the probabilility of that trigram being selected randomly from a text in that language, and stored in an external file. For example, the sequence "TZE" was found 37 times in the sample of 1,058,761 characters (1,058,759 trigrams) of German, giving a probability of occurrence of about 3.49 E -5 (3.49 times 10 to the power -5). To prevent trigrams which were not found in the million word sample being regarded as impossible in that language, we set a default value of 0.5 divided by the number of trigrams in the sample as the probability of that trigram. In our system, queries to be analysed had to consist of at least one character. We automatically assigned initial and terminal blank characters to each query.
As an example, the overall probability of encountering the sequence _KATZE_ (where the underscore denotes a space character) was found by multiplying the five constituent trigram probabilities for each language (shown in Table 1), and multiplying them together to give an overall probability. Since the overall probability was much greater for German than the other two languages, KATZE is more likely to be a German word than an English or French one. Using this method, each of the first 100000 queries in the original search logs was annotated with the most likely language of that query. Only the query field in the search logs was required for this process.

Table 1. The five constituent trigrams in "_KATZE_", their individual probabilities and their product for three languages.

Trigram	_KA	KAT	ATZ	TZE	ZE_	Overall Probability
English	1.23 E-5	4.72 E-7	4.72 E-7	3.49 E-5	1.70 E-5	1.63 E-27
French	5.00 E-7	5.00 E-7	5.00 E-6	5.00 E-7	1.01 E-5	6.31 E-32
German	5.64 E-4	1.41 E-4	1.41 E-4	1.41 E-4	1.12 E-4	1.77 E-19

Other methods for automatic language identification include the text compression technique of Benedetto et al. (2002), which is said to be effective with as few as 20 characters. However, the technique can not be performed using standard programming techniques. A more common technique for automatic language identification is counting the number of language-specific stop words in each text, but this needs longer texts to work with.

6 The Languages of the Individual Queries

According to the automatic language recognition program, the languages of the 100000 individual queries were (from the set of 11 Europarl languages) were as shown in Table 2. English was the most commonly used query

language, followed by Italian, but as discussed in section 8, many of the "Italian" queries may in fact have been in Latin. In almost 10% of the query lines, no text was submitted.

Table 2. Languages of the first 100000 queries

Language	Use per 100000 (including null)	Percentage (excluding null)
English	26732	29.69
Italian	12007	13.38
German	11045	12.27
NULL	9975	
French	8605	9.56
Dutch	7056	7.84
Spanish	6726	7.47
Finnish	5051	5.61
Portuguese	4826	5.36
Swedish	4226	4.69
Greek	9	0.00

7 Do Users Change Language within a Session?

The purpose of this experiment was to use the search logs annotated with the most likely language of the query to find whether users tended to stick with one language throughout a search session, or whether they tended to change languages in mid-session as part of the query reformulation process. If they did seem to change languages mid-session, which languages did they most commonly change from and to? The time-ordered set of query logs was scanned, and each time the session ID did not match the session ID of the previous query, it was assumed that a new query had begun. Otherwise, if the previous and current session IDs were the same, entry [previous_state][current_state] in the matrix was incremented by 1. The results are shown in Table 3, where the earlier states are on the vertical axis, while the later states are found on the horizontal axis. "new" denotes the start or end of a session, and "null" indicates no query was submitted at this stage.

Table 3. Query language used in consecutive stages of query reformulation

	New	null	DAN	GER	GRE	SPA	FIN	FRE	ITA	DUT	POR	SWE	ENG
New	0	5990	1649	5306	4	3212	2384	4021	5633	3118	2319	2111	12901
Null	4838	3557	73	207	0	116	101	155	182	127	91	59	468
DAN	1734	14	1880	15	0	9	5	8	13	16	2	15	31
GER	5462	48	19	5287	0	21	19	20	36	37	12	15	69
GRE	3	0	0	0	5	0	0	0	1	0	0	0	0
SPA	3287	26	6	19	0	3218	11	27	37	20	16	14	45
FIN	2452	34	14	12	0	11	2412	17	24	13	13	11	38
FRE	4129	35	13	22	0	28	14	4224	38	22	9	9	62
ITA	5765	47	12	30	0	24	22	46	5916	23	17	13	92
DUT	3184	43	15	54	0	18	18	14	19	3619	10	14	48
POR	2401	18	6	8	0	13	7	12	24	8	2303	1	25
SWE	2133	21	14	22	0	13	12	11	13	18	1	1936	32
ENG	13259	142	41	63	0	43	46	50	71	35	33	28	12921

The most common language for the first query of a session was English, followed by German. In the vast majority of cases, as shown by the high values on the principal diagonal, users having submitted a query in one language tended to use the same language for the next query. If users did change language mid-session, there was a slight tendency to change into English, shown by the slightly higher values in the final column. The other values in the matrix may represent a "noise floor" due to incorrect assignments by the automatic language identifier.

Table 3 can be transformed into a Markov model, by dividing each entry by the row total. In a Markov model, events such as the submission of a query in a particular language are regarded as states (denoted by the country

codes in Table 3), as are the start or end of session (denoted "new"), and submission of no query (denoted "null").

In a bigram model, the likelihood of a event occurring depends on the state of the system. For example (see again Table 4, where rows refer to current states and columns refer to next states), if we are in the state of a new session, there is an 8% chance that the first query will be in French; if the current query is in Italian, there is a 49% chance that the next query will also be in Italian; if the current query is in English, there is a 50% chance that at the next state the session will close. In higher order Markov models, the likelihood of an event next occurring depends not only on the current state, but on a number of previous states. In fact, the trigram probabilities for each language used by the automatic language identification program described in section 4.1 is also a Markov model, where the states are individual characters or spaces. The probability of the trigram "KAT" might be described as the likelihood of T coming next, given that the current state is A and the previous state was K.

Table 4. Markov model for the sequence of query languages in the search logs

	new	**null**	**DAN**	**GER**	**GRE**	**SPA**	**FIN**	**FRE**	**ITA**	**DUT**	**POR**	**SWE**	**ENG**
New	.00	.12	.03	.11	.00	.07	.05	.08	.12	.06	.05	.04	.27
Null	.49	.36	.01	.02	.00	.01	.01	.02	.02	.01	.01	.01	.05
DAN	.46	.00	.50	.00	.00	.00	.00	.00	.00	.00	.00	.00	.01
GER	.49	.00	.00	.48	.00	.00	.00	.00	.00	.00	.00	.00	.01
GRE	.33	.00	.00	.00	.56	.00	.00	.00	.11	.00	.00	.00	.00
SPA	.49	.00	.00	.00	.00	.48	.00	.00	.01	.00	.00	.00	.01
FIN	.49	.01	.00	.00	.00	.00	.48	.00	.00	.00	.00	.00	.01
FRE	.48	.00	.00	.00	.00	.00	.00	.49	.00	.00	.00	.00	.01
ITA	.48	.00	.00	.00	.00	.00	.00	.00	.49	.00	.00	.00	.01
DUT	.45	.01	.00	.01	.00	.00	.00	.00	.00	.51	.00	.00	.01
POR	.50	.00	.00	.00	.00	.00	.00	.00	.00	.00	.48	.00	.01
SWE	.50	.00	.00	.01	.00	.00	.00	.00	.00	.00	.00	.46	.01
ENG	.50	.01	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.48

A variant of this program was written to calculate the proportion of sessions in which more than one language was used. Here "null", or no query submitted, was not considered as a language. The relative proportions of sessions consisting of zero, one or more than one language are shown in Table 5.

Table 5. Proportion of sessions consisting of zero, one or more than one language

Number of Languages Used	**Number of Sessions**	**Percentage**
0	4701	9.66
1	42354	87.07
>1	1592	3.27

8 Correlation between the Portal Interface Language and the Query Language

The purpose of this experiment was to answer the question: Do users tend to submit queries in the same language as the interface they have chosen? First, the number of sessions conducted in each interface was collated, as shown in Table 6.

Table 6. Sessions grouped by language of the interface

Language code	**Frequency**	**Percentage**	**Language code**	**Frequency**	**Percentage**
En	41109	84.50	**Sr**	84	0.17
Pl	2012	4.14	**Sk**	79	0.16
Fr	1993	4.10	**Cs**	75	0.15
De	1145	2.35	**Nl**	55	0.11
It	511	1.05	**Lt**	41	0.08
Pt	353	0.73	**El**	28	0.06
Lv	346	0.71	**Fi**	13	0.03

Sl	340	0.70	**---**	10	0.02
Hu	245	0.50	**Da**	9	0.02
Hr	103	0.21	**Mt**	5	0.01
Et	91	0.19			

The matrix in Table 7 was then generated by reading in each line of the annotated query logs in turn, reading off both the language of the interface and the language of the query and incrementing the entry [interface_language][query_language] by 1. The rows show the interface language, while the columns show the query language. For the most popular interfaces, German, French, Italian, Dutch, Portuguese and English, the most common query language was the language of the interface, followed by English. When the interface was English, the most common query language apart from English itself was Italian. This may in fact be due to fact that users were searching for documents with Latin titles. Latin is not included in the Europarl corpus, so our automatic language identifier in cases of Latin queries may have returned the most similar language, Italian. For example, the Latin query "Commentaria in Psalmos Davidicos" was returned as Italian. Another interesting finding was that since there was no Spanish interface, many users wishing to submit Spanish queries used the interface of the closest language, Portuguese.

Table 7. Cross-tabulation for the choice of interface language and query submitted

	null	**DAN**	**GER**	**GRE**	**SPA**	**FIN**	**FRE**	**ITA**	**DUT**	**POR**	**SWE**	**ENG**
Null	0	0	0	0	0	0	0	0	0	0	0	0
DAN	0	7	2	0	0	0	7	0	0	0	0	0
GER	380	97	360	3	167	84	98	203	243	115	100	535
GRE	15	0	0	0	4	1	2	12	0	4	12	4
SPA	0	0	0	0	0	0	0	0	0	0	0	0
FIN	0	0	8	0	4	0	0	0	0	3	0	5
FRE	561	115	320	0	265	266	1021	406	130	148	66	789
ITA	121	14	30	0	142	65	88	318	22	29	8	152
DUT	16	0	14	0	9	0	14	9	210	2	4	43
POR	106	11	19	0	222	19	36	137	64	73	5	51
SWE	4	1	26	0	0	5	0	6	3	23	25	26
ENG	7856	3316	9109	6	5496	3970	6985	10107	6014	4126	3650	23892

9 Conclusion

We have developed a search engine, where previously accessed documents are indexed by all the search terms, derived from search logs, that have ever been submitted in the same sessions as those in which that document was downloaded. New queries are matched against the old query terms in the indexes, and documents are ranked by the degree of match between their index terms and the new query. In order to learn about multilingual searching behaviour, we have performed automatic language identification using trigram frequencies at the time the queries are indexed. We were then able to record the sequence of languages used, and build a Markov model to store the relative frequencies of sequences of query languages.

Acknowledgements

This work was supported by the EU-Funded VITALAS project (project number FP6-045389)
http://vitalas.ercim.org

References

Benedetto, D.; Caglioti, E.; Loreto, V (2002): Language Trees and Zipping. In: Physical Review Letters 4.

Cui, H.; Wen, J-R; Nie, J-Y.; Ma, W-Y. (2003): Query Expansion by Mining User Logs. In: IEEE Transactions on Knowledge and Data Engineering, 15(4), pp. 829-839.

Europarl Parallel Corpus. www.statmt.org/europarl

Hoi, C-H.; Lyu, M. R. (2004): A Novel Log-Based Relevance Feedback Technique in Content-Based Image Retrieval. In: ACM Multimedia, pp. 24-31.
Souter D; Churcher G; Hayes, G.; et al (2004) : Natural Language Identification Using Corpus-Based Models. In: HERMES Journal of Linguistics 13, pp. 183-203.