Query Word Deletion Prediction

Rosie Jones
Overture Services
74 N. Pasadena Ave
Pasadena, CA, 91103 USA
rosie.jones@overture.com

Daniel C. Fain
Overture Services
74 N. Pasadena Ave
Pasadena, CA, 91103 USA
dan.fain@overture.com

ABSTRACT

Web search query logs contain traces of users' search modifications. One strategy users employ is deleting terms, presumably to obtain greater coverage. It is useful to model and automate term deletion when arbitrary searches are conjunctively matched against a small hand-constructed collection, such as a hand-built hierarchy, or collection of high-quality pages matched with key phrases. Queries with no matches can have words deleted till a match is obtained. We provide algorithms which perform substantially better than the baseline in predicting which word should be deleted from a reformulated query, for increasing query coverage in the context of web search on small high-quality collections.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation*

General Terms

Experimentation

Keywords

Web search, query reformulation, query modeling

1. RELATED WORK

Spink et al. [4] found that reformulations of queries constituted 40% to 52% of queries. Other research using web search query logs has sought to characterize the individual queries [3], or to use the clickthroughs to optimize retrieval [1] performance. Work on query expansion looks at queries in isolation, and attempts to increase recall by expanding the query [5]. Lau and Horvitz [2] learn to predict general web search behaviors from query logs annotated with search strategies and goals, ignoring the words in the query. This work differs in that experimental results are compared to the query reformulations made by real users, with term deletion examined as a specific reformulation strategy.

SIGIR'03, July 28–August 1, 2003, Toronto, Canada.
ACM 1-58113-646-3/03/0007.

Reformulation	%	Example
non-rewrite	53.2%	mic amps → create taxi
insertions	9.1%	game codes → video game codes
substitutions	8.7%	john wayne bust → john wayne statue
spell correction	7.0%	real eastate → real estate
mixture	6.2%	huston's restaurant → houston's
deletions	**5.0%**	**skateboarding pics → skateboarding**
specialization	4.6%	jobs → marine employment
generalization	3.2%	gm reabtes → show me all the current auto rebates
other	2.4%	thansgiving → dia de accionde gracias
(de)pluralization	0.5%	video cards → video card

Table 1: Summary of query reformulations. Non-rewrites include pairs of queries on the same topic, but not directly related; pairs of queries which correspond to navigation; and pairs which are apparently unrelated.

2. QUERY LOG DATA

We use complete logs of user web accesses from April through December 2002, recorded on client-side software provided by an ISP. In our logs there are on average 693,000 users accessing web pages per day ($\sigma = 218$). An average user makes 79 web accesses per day ($\sigma = 116$), with an average of 3.6 web searches per day ($\sigma = 11$). A *candidate reformulation* or *query pair* is a pair of successive queries to any search engines issued by a single user on a single day. We collapse repeated searches for the same terms, as well as query pairs repeated by the same user on the same day. This results in 600,000 query pairs daily. To examine the frequency and type of query reformulation, we chose 1236 query pairs uniformly at random and labeled them by hand. The breakdown by labels is summarized in Table 1.

3. DELETION PREDICTION

In our data there are 5,514,971 instances of query pairs in which a single word is deleted. The average length of these queries is 3.07 ($\sigma = 1.17$) words, so we would expect deleting a random word to match the actual word deleted 33% of the time. A model which predicts which word to delete from a query can help a user to refine an overly specific query to one which has more coverage. Most query reformulation research focuses on query *expansion* (eg. [5]). Our research on query term deletion prediction is novel, and applicable to at least 5% of all web search query reformulations. We use as training data the single-word deletion data from April to December 2002. The test set is the first 2000 instances of

Deleted Word	Num queries	Num queries deleted from	$\hat{P}_{joint}$	$\hat{P}_{conditional}$
megaherzt	1	1	1.8e-7	1
reputable	10	9	1.6e-6	0.9
biography	7626	6228	0.001	0.82
birthplace	69	55	9.97e-6	0.80
orphaned	12	9	1.6e-6	0.75
playwright	34	22	3.99e-6	0.65
pictures	92901	55037	0.01	0.59
download	47764	27558	0.005	0.58
nude	82809	43603	0.008	0.53
free	188996	91226	0.02	0.48
the	113515	43464	0.008	0.38
and	73169	23642	0.004	0.32
in	81414	23298	0.004	0.29
of	129347	29613	0.005	0.23
osiris	177	28	5.0e-6	0.15
yangtze	19	3	5.4e-7	0.15
tupperware	755	45	8.2e-6	0.06
yamahar6	1	0	0	0

Table 2: A sample of words and corpus-based estimates of $P(deletion|inQuery)$. The top scoring words are generally typos and misspellings.

single-word deletion from January 1st, 2003.

In *leftmost deletion* we predict that the leftmost word of a query is deleted. This scheme is intuitive, if queries are mostly in English, and consist of well-formed noun-phrases, with optional adjectives appearing on the left. In addition, it is applicable to rare queries, for which we have no information about the likelihood of deletion of individual words. *Rightmost deletion* is analogous to leftmost deletion and makes sense if queries are built up from left to right, with the user inputting the most important terms first, then adding less important terms towards the end.

Joint probability deletion is a naive statistical model of word deletion. For example, "free" is deleted from a query 91,226 times, so over all 5,514,971 examples of single word deletion we estimate the joint probability of deletion to be 1.65%. We predict that the word in the query with the highest joint probability is deleted. This model has the disadvantage of preferring frequently deleted words, over words which are *consistently* deleted.

Conditional deletion probability overcomes the shortcomings of joint probability deletion, by factoring in the frequency of a word, calculating $P(delete(\omega)|contains(\alpha,\omega))$.

A sample of words and their joint and conditional probabilities are shown in Table 2. Note that the most frequently deleted words are not among the highest conditional probability of deletion. Note also that words at the extreme ends of the conditional probability scale are generally singleton typos and misspellings. Between these extremes, we find more modifiers with high probability of deletion, such as "reputable", "biography" and "birthplace", and proper names with low probability of deletion, such as "osiris", "yangtze", and "tupperware". The latter presumably form the core of the concept the user is searching for, with modifiers deleted in the hopes of improving search coverage.

In *history-based deletion prediction* we look in the training data for instances of the query. We then compare frequency of deletion for each term among the matching instances of the query. For the query "desktop computers", we have 86 instances of single-word deletion in the query history. Of these, in 74 cases, the word "desktop" was deleted, and in 12 cases the word "computers" was deleted. We propose "desktop" as the most likely term to be deleted. In our test set, 494 cases matched previous queries in the logs. In order to obtain a broad-coverage deletion predictor, we need to provide an alternative scheme for the unmatched cases. We combine history-based deletion prediction with a backoff to rightmost deletion, and conditional probability deletion.

Condition	Correct	Incorrect	Correct %
random	709	1291	35.4%
leftmost	589	1411	29.5%
rightmost	1012	988	50.6%
joint probability	936	1064	46.8%
conditional probability	1056	944	52.8%
conditional prob.; rightmost	1054	946	52.7%
history; rightmost backoff	1109	891	55.5%
history; conditional prob.	1110	890	55.5%

Table 3: Deletion prediction on 2000 queries with single-word deletions from test set. The standard error for all cases is between 0.9% and 1.1%. Backoff schemes are shown after the semi-colon, where no backoff scheme is used, alphabetical ordering is used to resolve ties.

4. DELETION PREDICTION RESULTS

Table 3 shows the results of these deletion prediction schemes on our test set. Rightmost deletion is accurate in 50.6% of cases, much better than deleting a random word, which has accuracy of 35.4%. Conditional probability predicts the correct word in nearly 53% of cases. On the 25% of cases where it is applicable, history-based deletion is very accurate, predicting the correct word to be deleted in 78% of cases. When we combine history-based deletion with rightmost deletion, the overall accuracy is 55.5%, significantly better than either conditional probability or rightmost deletion alone. The efficacy of rightmost deletion in comparison to leftmost deletion suggests that the model of users building up queries from left to right is more accurate than the model of queries as noun-phrases with optional adjectives on the left. These results show that we are able to model the deletions users perform on their queries. This can help increase query coverage in the context of web search on small high-quality collections.

5. REFERENCES

[1] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2002.

[2] T. Lau and E. Horvitz. Patterns of search: Analyzing and modeling web query refinement. In A. Press, editor, *Proceedings of the Seventh International Conference on User Modeling*, 1999.

[3] C. Silverstein, M. Henzinger, H. Marais, and M. Moricz. Analysis of a very large Altavista query log. Technical Report 1998-014, Digital SRC, 1998.

[4] A. Spink, B. J. Jansen, D. Wolfram, and T. Saracevic. From e-sex to e-commerce: Web search changes. *IEEE Computer*, 35(3):107–109, 2002.

[5] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *Proceedings of SIGIR*, pages 4–11, 1996.