The Wild Thing Goes Local

Kenneth Church
Microsoft
One Microsoft Way
Redmond, WA
+1 (425) 7056457
church@microsoft.com

Bo Thiesson
Microsoft
One Microsoft Way
Redmond, WA
+1 (425) 7039362
thiesson@microsoft.com

ABSTRACT

Suppose you are on a mobile device with no keyboard (e.g., a cell phone) and you want to perform a "near me" search. Where is the nearest pizza? How do you enter queries quickly? T9? The Wild Thing encourages users to enter patterns with implicit and explicit wild cards (regular expressions). The search engine uses Microsoft Local Live logs to find the most likely queries for a particular location. For example, 7#6 is short-hand for the regular expression: /^[PQRS].*[ ][MNO].*/, which matches "post office" in many places (but "Space Needle" in Seattle). Some queries are more local than others. Pizza is likely everywhere, whereas "Boeing Company," is very likely in Seattle and Chicago, moderately likely nearby, and somewhat likely elsewhere. Smoothing is important. Not every query is observed everywhere.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval] – *Retrieval Models*

General Terms: Algorithms

Keywords: Near me, Wild Thing, Local Search.

1. The Demo

The Wild Thing [2] inputs a regular expression query, *q*, and a language model, *LM* (a list of queries and their popularities in the search logs), and outputs the k-best (most popular) matches. We could do this with grep, though it is faster to use an index [3]:

```
grep q LM | sort -nr | head
```

The local version of the Wild Thing takes a location as an additional input. The task is to find the k-best *nearby* matches.

For the local version, we use the logs from a local search service (Local Live) instead of the logs from a general web search service (Live). Local queries are different from standard web queries. "Pizza" is relatively popular in the local logs, whereas queries for web services (mail, news, shopping and adult entertainment) are more popular in general web logs.

Figure 1: p → Pearl Harbor (1st choice) in Honolulu.

The demo finds different matches in different locations. "P" matches "Pearl Harbor" in Honolulu, but not in Providence. Most single letter queries mean different things in different places: F → Ford (in Detroit), but Ferry (in New London). Similar comments hold for initials: B C → British Columbia, Boeing Company, Baptist Church and Bible College, depending on location. Many other patterns show interesting variations by location: *beach, *high, *school, *univ, *hospital, *airport, *river.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
ACM 978-1-59593-597-7/07/0007.

2. Smoothing

Smoothing is important, both for computational and statistical reasons. We can't materialize (or estimate) the probability of every query in every location.

The training data (local logs) consist of a sequence of queries and locations. We build a kd-tree[1], splitting the world up into 32k tiles, where each tile has an equal number of queries.

Initially, counts for a query are assigned to the leaves. The smoothing process moves counts up the tree unless there is strong evidence for locality (a query at a node has significantly more counts than the same query at the sibling node). Counts pushed up the tree contribute to larger geographies (all leaves they dominate). There is a single free parameter, the significance level (*p*), that determines the level of smoothing.

A Wild Thing index[3] is constructed for each node in the tree. Results for each index on the path from a leaf to the root are combined to produce the overall k-best matches near a location.

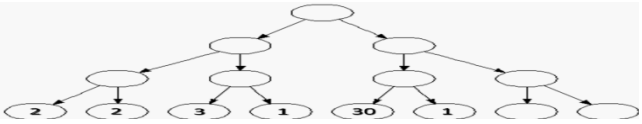

Figure 2: Counts are initially assigned to leaves of kd-tree.

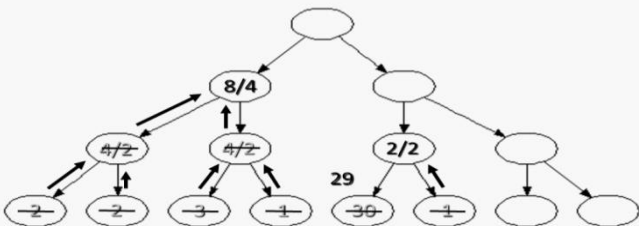

Figure 3: Counts flow up tree, unless significantly larger than sibling. 8/4 means 8 counts are evenly shared by 4 leaves.

3. REFERENCES

[1] Bentley, J. Multidimensional Binary Search Trees Used for Associative Searching, *CACM*, 18:9, 1975, pp. 509-517.

[2] Church, K., and Thiesson, B., The Wild Thing, *ACL*, 2005.

[3] Church, K., Thiesson, B. and Ragno, R., K-Best Suffix Arrays, *NAACL*, 2007.