Improving Similar Document Retrieval Using a Recursive Pseudo Relevance Feedback Strategy

Kyle Williams†, C. Lee Giles‡
†Information Sciences and Technology, ‡Computer Science and Engineering
The Pennsylvania State University, University Park, PA 16802, USA
kwilliams@psu.edu, giles@ist.psu.edu

ABSTRACT
We present a recursive pseudo relevance feedback strategy for improving retrieval performance in similarity search. The strategy recursively searches on search results returned for a given query and produces a tree that is used for ranking. Experiments on the Reuters 21578 and WebKB datasets show how the strategy leads to a significant improvement in similarity search performance.

1. INTRODUCTION
Finding similar files is a common use case for search in digital libraries, as in research paper recommendation [1] or near duplicate detection. One of the requirements in similarity search is that the query documents have features in common with the authored documents, which may not always be the case even when documents are semantically similar [2]. This problem is commonly known as feature mismatch and relevance feedback methods based on user feedback or the top k results have been developed for query reformulation. We present a pseudo relevance feedback search strategy for similarity search. Unlike most work in relevance feedback where query reformulation is used based on a set of documents, we instead generate a set of new queries, where each new query is based on one of the top k returned search results. We perform this recursively on each search result and the output of our search is a tree, which we use for ranking.

2. RECURSIVE SEARCH AND RANKING
For an initial query \( Q \) that returns a set of results \( \mathbb{R} \), the strategy involves recursively searching on the top \( k \in \mathbb{R} \) for some recursive depth \( d \) and then combining and ranking the results of all searches. The output of the recursive search process is a tree as shown in Figure 1. In the tree, a directed edge from a node \( r_1 \) to a node \( r_2 \) represents the fact that \( r_1 \) was used as a query to retrieve \( r_2 \). For instance, the query \( Q \) was used to retrieve \( r_{1,1} \) and \( r_{1,2} \); \( r_{1,1} \) was used to retrieve \( r_{2,1} \) and \( r_{2,2} \); \( r_{2,1} \) to retrieve \( r_{3,1} \), etc. We use this tree for ranking based on the intuition that if a document was retrieved using the recursive feedback mechanism and not by the initial query, then it does not have features in common with the initial query and thus its score should be penalized. We propose to penalize search results based on their tree distance from the query document and, by doing this, we aim to account for the problem of performing blind relevance feedback on search results that are not relevant.

Assume a scoring function exists \( \varphi(\cdot) \) exists that calculates the similarity between a query document \( q \) and a search result \( r \). We then define a set of ranking formulas \( \Psi(\varphi, T) \) that assign scores to documents based on both the similarity score \( \varphi \) and the search result tree \( T \) produced through the recursive search. Equations 1-5 represent a few simple formulas that are used in this study.

\[
\psi_{Flat}(\varphi(q, r), T) = \varphi(q, r)^{T \cdot \text{depth}(r)}
\]

\[
\psi_{Power}(\varphi(q, r), T) = \varphi(q, r)^{T \cdot \text{depth}(r)}
\]

\[
\psi_{Decay}(\varphi(q, r), T) = \varphi(q, r)\left(1 - \frac{1}{\text{depth}(r)}\right)^{1+\frac{T}{\text{depth}(r)}}
\]

\[
\psi_{Log}(\varphi(q, r), T) = \varphi(q, r)^{1+\log_{10}T \cdot \text{depth}(r)}
\]

\[
\psi_{Div}(\varphi(q, r), T) = \frac{\varphi(q, r)}{T \cdot \text{depth}(r)}
\]

The design of the ranking functions above is based on our intuition that the tree distance from the query document to a search result is correlated with relevance and that search results should be penalized for having higher tree distances. Thus, all of our ranking functions are designed to capture this in various ways by considering \( T \cdot \text{depth}(r) \), which is the depth in the tree \( T \) at which \( r \) occurs. The first ranking function, \( \psi_{Flat} \) (Equation 1), does not actually consider the
We gratefully acknowledge partial support by the National Science Foundation.

5. REFERENCES