Query Word Occurrence Patterns: A Retrieval based Analysis

Abstract

The problem of informational retrieval on the web increases in complexity and data size that needs to be managed. This increase in size causes a steep increase in the noise level accompanying a user’s search for a particular information need. To weed through this noise, we need effective retrieval strategies; many systems in use still employ the vector space model that computes the cosine similarity score between a query and a document. Though effective and efficient in many cases, it ignores the conceptual information that could be encoded in the flow of the query words or related concepts across a document. In this paper, we present retrieval strategies based on encoding the flow of the query concepts across documents and using this coding to re-rank retrieval results based on the standard vector space model. Though our preliminary results do not show an improvement, the analysis reveals that there could be some benefit involved in using the spatial context of words.

1 Introduction

Seeking relevant information on the World Wide Web is a task most of us undertake for the varied information needs that govern our lives. This need varies from the completely well-defined and well-known (for e.g. an article on Vietnam War) to the somewhat obscure and hidden. It is this vast range that makes the problem of effective retrieval on the scale of the internet really hard. All the countless search engines out there attempt to make this job easier for the user, removing the noise and clutter of billions of documents by allowing the user to type in a few keywords and filter the results. In the process presenting to the user a list of ranked results by some notion of relevance defined by the algorithm used.

Most search engines possibly use hundreds of heuristics and schemes to rank a search result. The most important underlying task is to analyze the text in the documents and associate the concept(s) represented by the queries to the concepts in the documents. A naive way of doing this is to let the query word explicitly define a concept and simply search for the query words in the documents. This can be achieved by representing a document as a vector of independently occurring unique words, namely the still widely used vector space model. The basic idea is to compose documents as high-dimensional vectors of words and their frequencies and then query vectors can be compared to the document vectors using the cosine measure. Many variants of these methods have been the dominant methods underlying various retrieval systems like SMART [2], the Okapi basic search system [3], IRIS [4] etc.

The most obvious problem with the vector space model is that any spatial information for the words appearing in the query and the documents is lost. It appears to us the flow of these words across a given document is an indicator of the concepts presented in the document besides simply the occurrence frequency of these words. This notion has been explored in [1] where the authors develop
the Fourier domain scoring method. The difference between FDS and other vector space similarity measures is that, rather than storing only the count of a frequency term per document, FDS stores a term signal. The term signal shows how the term is spread throughout the document. If the spectrum of the query term signals is compared, then according to the authors, we can observe which documents have a high occurrence of the query terms and which documents have the query terms appearing together. This information is obtained by comparing the magnitude and phase of the spectrum across different term signals, respectively.

In this paper we present a perhaps more intuitive model for capturing word flow across documents. A document consists of sentences and words comprise these sentences. The conceptual flow of a document is marked by the transitions between the sentences. To capture this conceptual flow in the words belonging to the sentences, it appears that the position of the words across sentences in a document could be employed. These positions could then be an important indicator of the underlying concept flow. In the FDS scheme use a less intuitive binning scheme to capture the word flow while using sentence occurrences seems like a better thing to do.

In the next section we present our document representation scheme and describe how polynomial interpolation can be used to encode spatial context information for the words. This information is then used in retrieval to re-rank the documents returned by the standard vector space model as implemented by lucene, a java based search engine API [5]. We compare our scheme to the vector space model using the recall-precision criteria and precision after \( N \) documents.

## 2 Document Representation

To obtain a spatial representation of query words occurring in a document, it is split into sentences based on the standard English language based delimiters. The sentences are ordered in the sequence that they occur in the document. The words are ordered in order of decreasing frequency and a matrix of size \( m \times n \) is constructed assuming the number of words in the query is \( m \) and the number of sentences in the document is \( n \). Let's suppose a sentence is an ordered set of words such that \( S_i = (w_j, ..., w_k) \) then the incidence matrix \( A \) for a given document and a query sequence \( q = (w_i, ..., w_l) \) is given by

\[
A_{ij} = \begin{cases} 
1 & \text{if } w_i \in S_j \\
0 & \text{otherwise}
\end{cases}
\]  

Thus in most cases the matrix \( A \) would be highly sparse. Note that the matrix provides a convenient coordinate chart to quantify the position of a query word in a given document based on a given sentence occurrence. A multiple occurrence in one sentence, though rare, would be counted as one. This gives us a grid representation of word flows across a document.

### 2.1 Pattern Comparison

A possible way of comparing occurrence of words in a given query in one document to another would be to directly compare the incidence matrices. This would be a very computationally intensive task and moreover there doesn’t appear to be an obvious way of comparing the occurrence patterns across the two matrices. However if we view the points of occurrence as parts of a global pattern then comparisons could be feasible. In this paper we propose the following two methodologies

1. **Polynomial Interpolation Comparison:** Each occurrence of a 1 for a query word has a coordinate position in the incidence matrix, that is, for example \( i \)th row and a \( j \)th column. If we were to view the incidence matrix as a coordinate plane, then the occurrence of the query words can be said to trace a particular shape or curve. This is done by joining each occurrence of a query sequence until all words are exhausted. An example is shown in figure 1. Hence, we note all the positions for the words occurring in the query and a given document and attempt to fit a polynomial curve through them. Firstly, the incidence matrix is ordered by decreasing frequency of occurrence of words and the sequence of sentences is the natural order in the document. To interpolate in a numerically stable manner we

\[ \text{2} \]
add a small jitter to both the \( x \) (sentences) and \( y \) (words) coordinates. The degree of the polynomial is a parameter that needs to be determined empirically. If, for e.g., the query word pattern for a document is extremely erratic or wave like, then a low degree polynomial won’t suffice. In this circumstance we choose a polynomial of degree, which is a close fraction of the total frequency of the query words occurring in the document. Since a document of high occurrence of query words will tend to be more erratic. In standard numerical computation a polynomial norm \[6\] for a given polynomial \( \sum_{i=0}^{n} a_i x^i \) is \( \sum_{i=0}^{n} a_i^2 \). To compare two different query word patterns encoded by polynomial coefficients \( P_1 = (a_0, \ldots, a_n) \) and \( P_2 = (b_0, \ldots, b_n) \), we compute \( \| P_1 - P_2 \|_2 \). Note that this comparison only makes sense between polynomials of the same degree. We use this comparison measure in our retrieval strategy described in the experiments section.

2. **Edit Distance:** The edit distance [7] between two strings is defined as the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. This provides us a natural measure to compare query word occurrences across documents. Since we have a binary string encoding the positions of the words across sentences, we can directly compare the pattern of occurrence of a 1. Though for large word patterns this can be computationally intensive. In the next section we describe the experimental setup for using these measures in the retrieval scenario.

3 Experiments

For the test data set we used a subset of TREC, the SJM, DOE and AP collections. We indexed these documents by basic Lucene Indexer using standard analyzer. Since we are employing sentence based processing, for runtime efficiency we could parse a document into sentences ad index each sentence as a document. This however will slow down the indexing stage. into We also used Porter Stemming Algorithm [8], to do stemming on documents as well as queries. For testing, we used queries 00101 to 00127 from task TREC-2. We basically used OR of all words in the concept field of each task as our query with no additional information from other fields. The algorithm used in retrieval follows the following steps:

1. Retrieve top 1000 files using Lucene for each query.
2. Find the Sentence-Word occurrence matrix for all words in the query in all sentences of every document as described in the previous section.
3. Calculate the coefficients of the polynomial curve that best fits the Sentence-Word space (described previously) for every document-query pair
4. Rerank documents based on the similarity of their patterns to original returned documents by Lucene.

We used different methods for the reranking section. Detailed reranking algorithm:
1. Result-list is empty  
2. Put every document in the remaining-docs-list 
3. Pick the best retrieved document by Lucene that is not already in the result-list  
4. Add this document to the end of the result-list  
5. Remove from remaining-docs-list  
6. Calculate all distance of other documents in the remaining-docs-list patterns to the current document.  
7. Sort them based on the distance  
8. Add the top K closest neighbors to this result-list  
9. Remove them from remaining-docs-list  
10. Repeat from step 3 if remaining doc list is not empty

This algorithm will continue and eventually put every document in the result-list. We used the general term distance and pattern in the algorithm because we can use different alternatives for calculating them. Our basic idea was to use coefficients of the polynomial curves that best describes the shape of points in Sentence-Word space as the representative of the pattern and Euclidian distance as the distance measure. We tried this approach and the results and results are shown in figure h1.
and h2. The best result was achieved by $K = 10$. We did not find great sensitivity to K for $K = 5$ to $K = 10$; performance is affected significantly for $K > 20$. We also calculated the Euclidean distance only between those documents that their patterns had the same number of coefficients which means could be modeled by the same type of curve. This might not be exactly the best way to do it after all, but resulted better than calculating the distance between every two pair, even with different highest degree. The main problem with comparing the basic best polynomial curve that fits the WS data is that its not stable to small changes. Figure 3 shows how at least one change in the WS data can change the best fit curve dramatically. The degree of the blue curve is 2 and for the red curve its 5. Two curves can explain almost the same data except for one point at (2.6) that the algorithm tries to find a curve that captures that point too and ends up with a degree 5 polynomial. The results are shown in figure 2(a) and 2(b).

Although we are not yet able to outperform the standard vector space model, it can be seen that working purely in polynomial coefficient space without using any word frequency information yields reasonable results. This is perhaps indicative of the importance of the flow of the patterns of the query concepts in a document. It is evident that the polynomial model is able to include relevant documents in the list just based on spatial information. However this also seems to imply that word frequency information cannot be ignored altogether. There could be a way of combining this spatial and frequency information into one measure for much better results, perhaps something similar to the technique used in [1]. Another issue in polynomial interpolation is that of translation and rotation variance. If two word patterns are essentially similar but occur across different locations in a document, the standard interpolation would not be able to capture this invariance. However, there exists literature on invariant methods that we intend to study. Also, as a sanity check we included randomly chosen documents in the above mentioned re-ranking algorithm. The $K$ nearest neighbors are randomly chosen and added to the list. The performance is shown in the plots.

Another approach that we tried for the distance measure was to calculate the average weighted distance from top $i$ to $i + m$ documents ($m$ docs) as the measure instead of distance from one document only; Where $i$ is the index of the best retrieved document in the remaining-docs-list. So we are comparing each document to the top m documents in the remaining-docs-list instead of just one giving more importance to the top one. This approach adding another free parameter, did not improve the performance significantly. Another alternative approach is use edit distance. We can use the same algorithm, using Sentence-Word raw data instead of the calculated polynomial coefficients as the pattern that represents document-query similarity, and use the edit distance of these patterns instead of the Euclidian distance of polynomial patterns.

4 Conclusion and Future work

In conclusion we feel that our results do indicate that spatial information is indeed correlated to concepts across different documents and if utilized in the correct manner can outperform vector space models in retrieving relevant documents. For the future we intend to find ways of combining frequency information with spatial information to come up with superior retrieval strategies. We also intend to test our claim more extensively on various other TREC competition data. Also since we have reduced query-document comparison to a pattern similarity problem, we could also utilize measures from the image processing literature. These could perhaps find application in the document clustering and classification domain. We are also currently running experiments on the edit-distance, however because of the computation intensive nature of the task the results are not done yet.

References


