Abstract. The paper presents a method for the usage of Boolean expressions for information retrieval based on Latent Semantic Indexing (LSI). The basic binary Boolean expressions such as OR, AND and NOT(AND-NOT) and their combinations have been implemented. The proposed method adds a new functionality to the classic LSI method capabilities to process user queries typed in natural language (such as English, Bulgarian or Russian) used in the "intelligent" search engines. This gives the user the opportunity of combining not only distinct words or phrases but also whole texts (documents) using all kinds of Boolean expressions. An evaluation of the implementations has been performed using a text collection of religious and sacred texts.

Introduction

The classic search engines give the user the opportunity to use keywords and/or Boolean expressions containing keywords. The "intelligent" search engines can process queries in natural language but do not permit the usage of Boolean expressions. We focus on the design of appropriate functions and mechanisms that will give the user the opportunity to combine free-form queries with Boolean operations in order to get better search results. The goal is achieved by combining the classic LSI algorithm with sophisticated implementation of the appropriate Boolean operations.

Latent Semantic Indexing

The Latent Semantic Indexing (LSI) is a powerful statistical technique for information retrieval. It is a two-stage process that consists of (see [2,3,4] for details): off-line construction of document index, and on-line respond to user queries.

The off-line part is the training part when LSI creates its index. First a large word-to-document matrix $X$ is constructed where the cell $(i,j)$ contains the occurrence frequency of the $i$-th word into the $j$-th document. After that, a singular value decomposition (SVD) is performed, which gives as a result three matrices $D$, $T$ (both orthogonal) and $S$ (diagonal), such that $X=DS^T$. Then all the three matrices are truncated in such a way that if we multiply the truncated ones $D'$, $S'$ and $T'$ we get a new matrix $X'$ which is the least-squares
best fit approximation of $X$. This results in the compression of the original space in a much smaller one where we have just a small number of significant factors (usually 50-400). Each document is then represented by a vector of low dimensionality (e.g. 100). It is possible to perform a sophisticated SVD, which speeds the process by directly finding the truncated matrices $D'$, $S'$ and $T'$ (see [1]).

The on-line part of our search engine (and of LSI) receives the query (pseudo-document) the user typed and finds its corresponding vector into the document space constructed by the off-line part using a standard LSI mechanism. Now we can measure the degree of similarity between the query and the indexed documents by simply calculating the cosine between their corresponding vectors. (see [5,6])

**Boolean operations**

Consider an e-commerce portal tracking the users’ purchases in order to offer them personalised advertisement: banners, etc. We can think of the purchases as query components and of the advertisement as new document in the same space. We need some kind of similarity function that will give us a measure of the similarity between our advertisements and the users “profile”. Let us define $d_1$, $d_2$, ..., $d_n$ as distances (in LSI sense) between the ad and the $n$ components of the query. The classic LSI algorithm calculates the cosines between the vectors to find the degree of their similarity. Most of the similarity measures for the Boolean operations we propose below are based on Euclidean distances, although we can use some other distances (angle, Manhattan distance, Chebushov distance, power distance, etc.). It is important to note that we must first normalise the vectors before calculating Euclidean distances. All Boolean operations proposed return a value between 0 and 1. Almost the same results can be obtained using the classic cosines but for some functions it is difficult to fit the values returned in the interval $[0,1]$. There are several similarity measures we have experimented with:
• **OR-similarity measure.** This measure depends only on the minimal distance between the document and the query components and has the following general representation: 
\[ S_{or} = f(\min(g(d_1), g(d_2), \ldots, g(d_n))) \]
where \( f(x) \) and \( g(x) \) are some one-argument functions.

In case we have more information for the query we can add weights to the query components and modify \( g(x) \) to \( g(x, \omega) \). So the formula is:
\[ S_{or} = f(\min(g(d_1, \omega_1), g(d_2, \omega_2), \ldots, g(d_n, \omega_n))) \]
OR similarity measure has well separated picks at the query components vectors. The similarity measures for two- and three-component query, \( f(x) = 1/(1+x) \), \( g(x) = x \) are shown on figure 1.

• **AND-similarity measure.** This measure depends only on the sum of distances between the document and the query components. This measure has the following general representation:
\[ S_{or} = f(g(d_1) + g(d_2) + \ldots + g(d_n))) \]
where \( f(x) \) and \( g(x) \) are some one-argument functions. Usually this measure can be thought of as a superposition of distinct similarity measures of the query components.

The similarity measure for two- and three-component query, \( f(x) = 1/(1+x) \), \( g(x) = x \) are shown on figure 2.

• **Combination of the previous two (AND-OR).** This similarity measure is a combination between the previous two.
\[ S_{and-or} = f(S_{and}, S_{or}) \]
We can use linear combination between \( S_{or} \) and \( S_{and} \) measures. \( S = k.S_{or} + (1-k).S_{and} \), where \( k \) is constant and \( 0 \leq k \leq 1 \).

Figure 3 shows the two- and three-component query results for \( k = 0.5 \). We still have two distinct parts like the OR-
similarity function but higher values in the middle region between them just like the AND-similarity function. Figure 4 is an example of a weighed combined similarity measure.

- **Binary NOT (AND-NOT)-similarity measure.** A common problem with the search engines is that they often return too much documents considered as “very similar” to the user query. In this case the user could specify what to exclude. So we have a composite query consisting of two natural language phrases: one saying what to include and the other one what to exclude. This leads to the creation of the binary NOT similarity measure. If a document is more similar to the exclude document text it will receive a similarity measure of 0. (see the second clause below) Otherwise we return a similarity measure between 0 and 1 that takes in account the distances to both documents. We can define the NOT-measure \( S_{not} = 1 - \frac{d_1}{1 + d_2}, \) when \( d_1<d_2, \) and \( S_{not} = 0, \) else. The result is shown on figure 5.

**Application to Religious and Sacred Texts**

The Boolean operations we propose have been tested on a document collection of religious and sacred texts we found at http://davidwiley.com/religion.html. We selected 196 different religious and sacred texts from 14 categories: apocrypha (acts, apocalypses, gospels, writings), Buddhism, Confucianism, Dead Sea scripts, The Egyptian Book of Dead, Sun Tzu: The Art of War, Zoroastrianism, The Bible (Old and New Testaments), The Quran and The Book of Mormons. The experiments were performed in a 30 dimensional space with a preliminary to SVD replacement of the frequencies in \( X \) (196 documents \( \times \) 11451 words) with logarithms (see [6] for detailed explanation). Fig. 6 illustrates the inter-document similarities given by the correlation matrix (196\( \times \)196), shown in 5 different colours for the
five correlation intervals: 87.5-100%, black colour; 75-87.5%, dark grey; 62.5-75%, grey; 50-62.5%, light grey; 0-50%, white.

The dark rectangles in the main diagonal show the high correlation between texts belonging to the same religion. For example: the black rectangle from the bottom right corner contains texts from the Book of Mormons. To the left and up on the main diagonal can be found the Quran, then the Old Testament (The Bible), then come the Zoroastrian texts, The New Testament (The Bible), the Sun Tzu’s Art of War, the Egyptian Book of the Dead and so forth. And the smooth rectangle in the upper left corner shows the relatively high similarity between all kinds of apocrypha present. We see for example that The Book of Mormons is more correlated to the New Testament than to The Old Testament.

The first class of experiments was “practical” and included composition of two or more different queries and their combination with Boolean operations we implemented.

The second class of “theoretical” experiments included the choice of two or more texts from the same space and performing queries using Boolean operations (OR, AND and NOT).

One of the most interesting experiments were those with the combined OR and AND similarity measure search using different values for the parameter $k$. Fig. 7 shows the distribution of the correlation coefficients returned by our search engine for the 196 documents using a text from the Sun Tzu’s Art of War and another one from the Egyptian Book of the Dead. We can see that the results vary which suggests that we can obtain quite different results by just tuning the parameter $k$. The tables below show an example of the Boolean operations at work.
Conclusion

We consider the technique of LSI to be very important in the future and continue our experiments with new kinds of similarity functions and their behaviour on different types of texts. Further work concerns study of the dependence of the best similarity function upon the text collection parameters.

References

[4] LSA 1990-99, see http://lsa.colorado.edu

Address: 1113, Sofia, 27 Acad. G. Bontchev Str., Rila Solutions