Information Retrieval Systems using an Associative Conceptual Space

Jan van den Berg
Dept. of Computer Science
Erasmus University Rotterdam
The Netherlands
Email: jvandenberg@few.eur.nl

Martijn Schuemie
Dept. Information systems
Delft University of Technology
The Netherlands
Email: m.j.schuemie@its.tudelft.nl

Abstract

An AI-based retrieval system inspired by the WEBSOM-algorithm is proposed. Contrary to the WEBSOM however, we introduce a system using only the index of every document. The knowledge extraction process results into a so-called Associative Conceptual Space where the words as found in the documents are organised using a Hebbian-type of (un)learning. Next, ‘concepts’ (i.e. word-clusters) are identified using the SOM-algorithm. Thereupon, each document is characterised by comparing the concepts found in it, to those present in the concept space. Applying the characterisations, all documents can be clustered such that semantically similar documents lie close together on a Self-Organising Map.

1. Introduction

The availability of huge collections of books, CD-ROMs, video movies, articles etc. in modern libraries or their respective depositories has prompted the creation of many intelligent search systems. These Information Retrieval systems (IR-systems) apply different levels of Artificial Intelligence (AI). Every IR-system needs a type of query to do its job. In the classical ‘boolean retrieval systems’ [1], the user can specify a query by summing up the words that should occur in the title or body of the documents. By comparing the words of the query to the data collected from the individual books, the search result is found and shown to the user. These systems show a well-known trade-off between precision and recall (see footnote 1): using more keywords in the query usually results into a higher precision and a lower recall, and

---

1 Comparing IR-systems is not easy because performance is a complicated notion. Besides inspecting issues like the degree to which the collection of items is covered and their user friendliness, IR-systems are mostly evaluated with respect to their recall (the part of the relevant information available that has actually been found) and to their precision (the part of the information found that is really relevant to the user) [1]. The notion of relevance is quite personal however, and therefore not unique which strongly complicates the evaluation.

2 Logic operators like ‘not’, ‘and’, and ‘or’ can sometimes be used to improve the query.

3 Data on the books are usually stored in an inverted index, containing the most important words with references to the documents in which they appear.
vice versa. Analysing this performance, we notice that boolean retrieval systems only
use individual keywords. They lack knowledge about the semantic relations between
these words and between the common underlying notions.
An approach to improve IR-systems is the `vector-space model’ [2] where documents
and queries are represented by vectors, each component of which corresponds to a
word. Every component’s value represents the number of times the word appears in
the text. Comparison of the query-vector to the document-vectors results in a set of
documents presented to the user. Another development is the use of ‘relevance
feedback’ [1] where the user can report to the system whether the documents found
are relevant to the user. The system can then try to find documents similar to these
relevant documents. However, still very little semantics is taken into account. Using a
‘thesaurus’ (a vocabulary where synonymous, semantically covering, or otherwise
related words are being collected) can solve this problem: the user’s query can be
augmented with words related by the thesaurus. The thesaurus is usually created
manually and their construction is therefore time-consuming.
Attempts have already been made to use automatically constructed domain-
knowledge. Two such systems, the WEBSOM [4] and the Aqua-browser [5], will be
described in the next section. Using these descriptions, a general architecture for an
IR-system will be derived. Thereupon, we introduce our ACR-WEBSOM IR-system.

2. Knowledge based IR-systems
The original WEBSOM-algorithm uses full-text documents as input. After having
removed all non-alphabetical characters and less frequent words, each remaining word
is represented by a unique n-dimensional real vector $x_i$ with random-number
components, where $i$ denotes the $i$-th word in the text. The relation between words is
determined using the average short context $X(i) = [E \{ x_{i-1} | x_i \}, \epsilon x_i, E \{ x_{i+1} | x_i \}]^T$,
where E denotes the estimated expectation value evaluated over the text corpus, and $\epsilon$
is a small scalar number (e.g., $\epsilon = 0.2$). The $X(i) \in \mathbb{R}^n$ constitute the input vectors to
a Self-Organising Map (SOM) called the word category map. Using this map, a
‘fingerprint’ of every book can be constructed consisting of a cluster histogram. These
histograms are used as input for a second SOM called the document map where
documents addressing similar topics are generally mapped close together.
IR-systems based on Connectionist Semantic Networks (CSN) [13] also start out
removing less frequent words. The remaining words are placed in a network where
each node represents a word and where the connection weights represent the strength
of the relations between words. Normally, these weights are based on word-co-
ocurrence: words that often appear close to each other are more strongly connected.
This CSN can then be made accessible to the user by means of a graphical interface
where words can be selected using mouse-clicks.
Both systems just briefly reviewed, show a new development in IR-systems: instead of
matching documents on a word-by-word basis using the words of the query, the
documents are analysed to find underlying concepts to which these words are related.
So, these systems attempt to compare the documents and the query given on a higher
level of abstraction.
In both approaches, it is assumed that full-text representations of all documents are available and small enough to be handled within a realistic timeframe. This is not always the case. In libraries for instance, there are usually many books of several hundred pages, not available in electronic format. To characterise each book, it would be preferable to use only a small part of the entire book such as the index.

3. ACS-WEBSOM

At first sight, an index may seem to hold little information on the meaning of the words occurring in the document. Otherwise, the words present in the index have been found significant enough to be mentioned. Moreover, part of the order in which they appear in the text can be reconstructed as depicted in figure 2. We could hereby make the assumption that if words occur on the same page they most likely are related. For the word-co-occurrence—method used by CSN-based IR-systems, the information available in the ‘converted index’ may suffice. However, books often contain a long array of concepts, so we suppose that the holistic coupling of documents to concepts as used by the WEBSOM performs better in describing the contents of the documents. We can use the WEBSOM (which demands vectors as input) by creating a conceptual space using word-co-occurrence, the so-dubbed Associative Conceptual Space (ACS). This space contains a set $X$ of $n$ vectors, $X = \{x_1, x_2, x_3, ..., x_{n-1}, x_n\}$, each vector $x_i$ consisting of $d$ components. Each such vector is one-to-one related to a word $w_i$ of the collection $W = \{w_1, w_2, w_3, ..., w_{n-1}, w_n\}$ of index words. We further apply the convention that the
\[ x_i(t+1) = x_i(t) + \eta(t) \frac{x_j(t) - x_i(t)}{|| x_j(t) - x_i(t) ||} \]

\( \eta(t) \) is the current learning-rate)

However, simply applying the given association-rule alone may easily lead to false associations where a related word that is moved closer, is also brought closer to an unrelated word. To counter this effect, an active forgetting rule is introduced:

\[ x_i(t+1) = x_i(t) - \lambda(\text{delta}) \frac{x_j(t) - x_i(t)}{\text{delta}} \]

where \( \text{delta} = || x_j(t) - x_i(t) || \). If the effect of the active forgetting is made mostly local, correct orderings in more remote parts of the conceptual space will remain intact. Therefore, \( \lambda(\text{delta}) \) should be decreasing over \( \text{delta} \). Some of the dynamics we could expect from a combination of learning through association and active forgetting, is shown in fig 3 (for more details, we refer to [10], [11]).

Figure 3: Active forgetting: the circles represent the repulse-behaviour.

We can form an ACS from book indices if these indices are converted as described earlier. To do so, the words in each converted index are concatenated into a text-string \( T_i \) of index-terms \( t_{ij} \), \( T_i = \{ t_{i1}, t_{i2}, t_{i3}, \ldots, t_{i m}, t_{i m+1}, t_{i m+2}, \ldots \} \) where \( i \) denotes the \( i \)-th book in the collection of \( k \) books. These strings in turn, are concatenated into one long string \( T* = \{ T_1, T_2, T_3, \ldots, T_{k-1}, T_k \} \). We define the vocabulary \( W \) as the set of (unique) words occurring in \( T* \). For each word \( t_* \) in \( T* \), we define a neighbourhood \( N_{t_*} \) with radius \( r \). Each element of the neighbourhood is activated in combination with \( t_* \).

Figure 4: Example of a neighbourhood with \( r=3 \) and the activation combinations made.

We now construct a conceptual space \( C_i \) by going through the entire text-string \( T* \) several times whilst also applying the forget-rule. Words that often occur have a lower informative value. This can also be taken into account yielding the association-rule:
Having applied the ACS-algorithm, the concept space $C_s$ contains a multidimensional knowledge structure. To use it, we can apply the WEBSOM-algorithm. First, we simply use the reference vectors of $C_s$ as training vectors for a SOM. This results in a two-dimensional map of the knowledge structure (figure 5). Similar to the original WEBSOM, this word category map is used to create a document map (figure 6).

\[ x(t+1) = x(t) + \eta(t) \frac{x(t) - x(t)}{||x(t) - x(t)||} \cdot \alpha(i,j) \]

\[ x(t+1) = x(t) - \eta(t) \frac{x(t) - x(t)}{||x(t) - x(t)||} \cdot \alpha(i,j) \]

\[ \alpha(i,j) = \frac{2 \cdot \text{freq}_i \cdot \text{freq}_j}{\text{freq}_i + \text{freq}_j} \]

\[ \text{freq}_i = \frac{|T_*|}{|W|} \]

\[ |.| \text{ returns the number of elements in a collection} \]

\[ \text{freq}_i = \text{frequency of word } i \]

\[ \text{freq}_j = \text{frequency of word } j \]

\[ \alpha(i,j) = \frac{2 \cdot \text{freq}_i \cdot \text{freq}_j}{\text{freq}_i + \text{freq}_j} \]

\[ \text{freq}_i = \frac{|T_*|}{|W|} \]

\[ |.| \text{ returns the number of elements in a collection} \]

4 In our experiments, a five-dimensional space tended to give the best ordering.
4. Conclusions and outlook

Experiments show that the ACS-algorithm is able to create an ordering recognisable by and acceptable to humans. Further evidence shows that the algorithm will create very similar orderings independent of the initial values for the reference vectors. The ACS-WEBSOM-algorithm compares documents based on concepts constructed from word-clusters instead of on individual words. It is expected to achieve a higher precision and recall than traditional IR-systems. The biggest problem at this moment is the time-complexity of the learning process. This is mainly due to the application of the forget rule, which takes up approximately 95% of the processing time.

Further research should first of all be focused on reducing the time-complexity of the algorithm. The possibility of using the ACS-WEBSOM-algorithm on full-text documents should also be investigated. A CSN-based IR-system using the converted index as mentioned in section 3, is another interesting approach to consider.

Bibliography:


5 Processing a document-set of 40 indices with a vocabulary of 2,439 words and a text-string of 20,621 words, took 8 hours and 10 minutes on a pentium-133 system.