An Incremental Approach to Text Representation, Categorization, and Retrieval

Paul O'Neil
Rome Laboratory

Abstract
Efficient and accurate information retrieval is a goal of just about everyone. Whether you are looking for information on the Internet, a book or article in the library, satellite imagery of missile silos, or a recipe for dinner, finding exactly what you want or need, even if you know exactly what you are looking for, can be an imposing and most difficult task. Many current techniques require an intimate understanding of the actual processes involved. The method presented in this paper provides an automatic representation of text data by vectors, which can then be manipulated to categorize and organize the data. Information can be retrieved without knowledge of the underlying process. The user can ask for information using normal discourse. This technology can also be applied to data mining and visualization.

Introduction
With the proliferation of personal computers, widespread Internet access, and massive growth of the World Wide Web, the amount of information available to the average person has increased quite dramatically over the past decade. Sifting through and finding information within this sometimes unstructured mass of data can be a daunting task. Much of the relevant information remains buried, and attempting to find what you need is a painstaking and laborious process, many times resulting in failure. Part of this problem can be traced to the methods most often used for querying information databases [1]. For example, a boolean search requires users to be familiar with how the search method works, so queries can be structured appropriately. If not, interesting and unfruitful results can occur, like finding an article on raisins instead of information pertaining to Bob Dole's presidential campaign. Or when searching the Web, many thousands of hits may be returned on a user query, with relevant information lying deep down in the list. Another part of the problem is how the information is represented. Simple, unstructured text information can be difficult to query, so alternate methods of representation need to be pursued. For these reasons, researchers are continuously looking for better and more efficient methods to retrieve information, in terms of both precision and recall.

The method introduced in this paper generates an automatic representation for free text information using vectors; then organization, categorization, and ultimately retrieval can be performed in a much more efficient and user-friendly manner. The next section describes the motivation for this method. The rest of the paper presents preliminary testing results, and a discussion of continued research and applications.

Motivation
Vector techniques can provide an automatic representation of free text information [2]. With little or no user intervention, vectors can be generated for a corpus of text, and then used as a means of organizing and retrieving data. The basic assumption is that word co-occurrence frequency within the data corresponds to the strength of the relation between words [3].

The vector representations for words are then combined to create a representation for groups of words (e.g., sentences, paragraphs, articles, groups of articles). Words and words groups are represented in the same dimension, so they can be compared against each other to make information retrieval more useful. Queries can be submitted in plain English, in the form of keywords, a simple question, or an entire article which discusses the subject matter being retrieved. These advantages make this form of representation enticing to developers of information retrieval systems.

One potential drawback of vector methods is difficulty in representing sentence structure [4, 5, 6]. For example, vectors representing the sentences Bill hit the ball and The ball hit Bill are exactly the same; but for many applications, this may not be important. An intelligence application which recognizes scenarios for prediction or identification to a human operator may only need to recognize generalizations about individual events [7]. The knowledge that there was a hitting and it involved both...
Bill and a ball may be enough for the event to be recognized. In fact, the loss in information may even add to the generalization powers of the system, allowing it to respond more fluidly when presented with unknown data.

The method proposed in this paper takes advantage of vector representations in a unique manner. A subset of words is chosen pseudo-randomly so that their distribution throughout the length of the training corpus is fairly uniform. Vectors for these words are randomly generated, and then used to calculate vectors for the remaining words in the corpus, based on context. A bootstrapping method is then invoked to automatically increase the dimension of the vectors. In this manner, calculations are minimized in the training process, and the dimension of the final vector representation is kept low.

Method

This section describes in detail the method of creating vectors for items in the text corpus. The corpus must first be processed before turning it over to the vector representation engine. This processing takes on two steps, stopping and stemming. Stopping involves removing words from the corpus which add little or no significant value to the meaning, and stemming reduces words to their roots, so that the same vector is generated for all words having the same root.

As stated earlier, we start with a small subset of n words (called seed words) from which the rest of the vectors are built. These words are selected so that their distribution is fairly uniform across the length of the corpus. As will be seen later, this will help avoid the problem of generating an abundance of zero vectors. Components for these vectors (of size n) are chosen at random from the interval [-1, 1], and the vectors are then normalized so they lie on the unit sphere in n-dimensional Euclidean space. A window size is chosen, reflecting the amount of context which will influence the vector calculation. For example, a window size of 5 indicates that two words on either side of the object word will influence the value of its vector. Vectors representing the remaining words in the corpus are generated using a modified version of Gallant’s algorithm [8]:

\[ C_i = \frac{1}{2} C_i^{\text{RW}} + \sum_{k \in \text{window}(i)} \left(1 - \frac{d(i,k)}{\alpha D'}\right) C_k \]

where \( C_i \) is the vector of the \( i \)th word, \( C_i^{\text{RW}} \) is the vector of the last word in the previous window, \( d(i,k) \) is the distance from word \( i \) to word \( k \), \( D' \) is the maximum of the \( d(i,k) \) over all words \( k \) in the current window, and \( \alpha \) is a weighting factor for the influence of \( D' \).

Once this process is complete, the initial seed word vectors are extracted. An additional set of \( m \) seed words is selected at random, with \( m \leq n \). The dimension of all of these vectors is then increased to \( m + n \) in the following manner. Suppose there were four initial seed words, and that four additional seed words are selected \((m = n = 4)\). Figure 1 shows the values for the initial seed vectors in matrix A, and matrix B contains the values (after the first training pass) for the additional seed word vectors. These vectors represent the relationship between the two sets of seed words, so extending the initial seed vectors to eight dimensions is simply a matter of filling in the remaining components with values representing the word relationships already calculated, as shown in matrix B'.

![Figure 1: Incrementing Dimension of Feature Word Vectors](image)

To extend the second set of seed word vectors to eight dimensions, the relationships among them need to be calculated. This is done by calculating the dot product of the vectors for corresponding words. In matrix C, for example, EF is the dot product value of vectors for word E and word F. Vectors for these eight seed words are now used as the basis for calculating vectors for the remaining words in the corpus, again using Gallant’s algorithm. This process of incrementing the dimension of the space and recalculating word vectors continues until the dimensionality is sufficient to encode relationships amongst the total number of words in the corpus.

Vectors for messages or articles are calculated by summing the vectors representing the individual words in the message or article, and then normalizing the result. Simple clustering of the messages is now based on the angle between the normalized vectors.

Results

A prototype system has been developed which generates representations based on the methodology described. Preliminary testing has been conducted, and the
results are presented below. Data for these experiments came from the TIPSTER Message Understanding and Text Retrieval Conferences (MUC and TREC, resp.). The TREC data was financial news extracted from the Wall Street Journal, and the MUC data was a group of articles discussing terrorist activities in South America. Figure 2 provides some more characteristics of the data set.

<table>
<thead>
<tr>
<th>Data Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td># of articles (messages)</td>
</tr>
<tr>
<td>total # of words in text corpus</td>
</tr>
<tr>
<td># of unique words after pre-processing</td>
</tr>
</tbody>
</table>

**Figure 2: Data Characteristics**

Forty seed words were selected at random, and vectors for these words were generated by choosing each component randomly from [-1, 1]. Vectors for the remaining words were calculated using Gallant’s algorithm with a window size of 5 and an $\alpha$ of 1.3. An additional set of forty words was chosen at random from the corpus, and using the method described earlier, the dimension of these two sets of vectors was increased to 80. Again word vectors for the rest of the corpus were calculated, and this process continued a couple more iterations.

**Figure 3: Message Clustering**

The data presented in this section reflects the results with vectors of 80 dimensions. Increasing the size of the vectors past this point had no substantial impact on the quality of data clustering and organization.

Figure 3 highlights the distances between messages in each domain. The darker shaded areas indicate a high dot product between the message vectors, and thus a shorter distance between them on the $n$-sphere.

As can be seen from the Figure 3, the MUC (terrorist) messages clustered extremely well, as did the Wall Street Journal (financial) messages. In addition, there was very little cross correlation of messages, and the reason for some of these correlations will be discussed shortly.

Using a Kohonen Self Organizing Map (SOM), further cluster analysis was performed on the data. Figure 4 shows the organization of data in the SOM, and displays the number of MUC (M) and WSJ (W) messages located at each unit. The shaded areas define the clustering that has taken place within the data.

**Figure 4: Self Organizing Map Results**

In unit (11, 4), there is one article from the Wall Street Journal sitting amongst the 46 MUC terrorist messages. A closer look reveals the reason for this. Below is an excerpt from the subject article, WSJ0179:

William L. Webster argued the "right to life" after receiving a death threat himself. ... Extra security surrounded him that day, ...

Now the question is whether that great victory will help translate into a great springboard -- into the governor's office. ... [he] has become identified with the most explosive issues of contemporary politics: The right to live and the right to die.

"There are people who say Bill Webster is trying to play God," he says in his conference room here. ... Attorney General Webster and his office have represented Missouri in three life-and-death U.S. Supreme Court cases within the past two years. ... In a second ruling last year the court, again upholding Missouri, ruled that the execution of a 16-year-old murderer would not be cruel and unusual punishment.
Notice there are discussions of a death threat, right to die, and playing God, all in regard to the politician Bill Webster, who may be running for governor. A look at some of the messages in the cluster around this one gives insight into its placement at this location.

<table>
<thead>
<tr>
<th>ID</th>
<th>DotPr</th>
<th>Subject of Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC0150</td>
<td>0.9800</td>
<td>Bomb threat at U.S. embassy</td>
</tr>
<tr>
<td>MUC0053</td>
<td>0.9745</td>
<td>Murder plot and death threat against government official</td>
</tr>
<tr>
<td>MUC0118</td>
<td>0.9730</td>
<td>Bombing of official's home</td>
</tr>
<tr>
<td>MUC0125</td>
<td>0.9681</td>
<td>Murder of government officials and car bomb explosion</td>
</tr>
<tr>
<td>MUC0022</td>
<td>0.9652</td>
<td>Priests murdered, human rights violations</td>
</tr>
<tr>
<td>MUC0121</td>
<td>0.9623</td>
<td>Bombing threats related to political elections</td>
</tr>
<tr>
<td>MUC0071</td>
<td>0.9617</td>
<td>Murder threats related to political elections</td>
</tr>
<tr>
<td>MUC0059</td>
<td>0.9560</td>
<td>Government officials killed</td>
</tr>
<tr>
<td>MUC0111</td>
<td>0.9560</td>
<td>Human rights violations, murder and kidnapping of politicians</td>
</tr>
<tr>
<td>WSJ0140</td>
<td>0.2445</td>
<td>Financial status of electrical engineering company</td>
</tr>
</tbody>
</table>

**Figure 5: Cluster near WSJ0179**

We can see in Figure 5 that many of these articles are about political figures who have received death threats or have been murdered, and some discuss political elections and human rights violations. The article WSJ0140 is the one farthest from WSJ0179, implying they are not related to each other. As can be seen from their subject matter, this is indeed the case.

The text representation and subsequent organization not only did a good job of separating the financial data from the terrorist data, but it also found more subtle differences within each of the two domains. For example, in Figure 4, the cluster represented by the units (10, 5), (11, 4), and (11, 5) is concerned with terrorist attacks on priests, the general population, and some political figures. The cluster in units (11, 2) and (12, 2) is more related to terrorist attacks on military personnel, and fighting between terrorist groups. These distinctions are important for both information retrieval and organization.

**Conclusions**

The impact of strong and efficient categorization and organization of information cannot be overstated. Successfully representing information with vectors in n-dimensional Euclidean space provides a useful mechanism for retrieving and browsing data.

The method described in this paper provides a robust and efficient means of creating vector representations for words and groups of words. Since words and words groups are represented in the same dimension, they can be compared against each other to make information retrieval more effective and user-friendly.

This system is being developed further, including integration with a neural network designed to perform assessment of temporal patterns. The output of this text representation method is ideally suited as input to a neural net. Thus free text information can now be used by planners and analysts to increase situational awareness, and make reporting more accurate by taking advantage of additional information that was previously only useable through a manual and time-consuming process.

**References**


