**Assignment No 1**

**AIM: Implementation of Conflation Algorithm.**

**OBJECTIVE:** To study

* Information Retrieval.
* Components of Information Retrieval System.
* Conflation Algorithm.

**THEORY:**

**Introduction to Information Retrieval**

In today’s information explosion era, increase in demand for quicker dissemination of

information, from contents stored in a variety of forms requires speedy search and timely retrieval. The values of documents are measured according to the information it contains but they are proved useless until the stored information is brought out for use by the readers. This may be either by subject analysis or representation of the terms through symbols. It has always been the need of the scholars and the lingering turmoil in the minds of library organisers, to suitably facilitate the extraction of the contents expeditiously and exhaustively that has brought forward the concept of information retrieval.

**Meaning & Definition:**

Calvin Mooers coined the term information retrieval in 1950. In the context of library and information science, we mean to get back information, which is, in a way, hidden, from normal sight or vision. According to

J.H. Shera: It is, "the process of locating and selecting data, relevant to a given requirement."

Calvin Mooers: "Searching and retrieval of information from storage, according to specification by subject."

**Functions:**

The major functions that constitute an information retrieval system, comprises of: Acquisition, Analysis, Representation of information, Organisation of the indexes, Matching, Retrieving, Readjustment and Feedback

**Components of Information Retrieval System:**

A study of the functions of IRS brings forth some of the essential components that constitute the proper functioning of the system. According to Lancaster, an information retrieval system consists of six basic subsystems. They are as follows:

1. The document selection subsystem
2. The indexing subsystem
3. The vocabulary subsystem
4. The searching subsystem
5. The user-system interface
6. The marching subsystem

All the above subsystems may be grouped under two groups' subject/content analysis and search strategy. Subject or content analysis includes the task of analysis, organisation and storage of information. Search strategy includes analysis of user queries, creation of search formula and the actual searching.

**Conflation Algorithm:**

Ultimately one would like to develop a text processing system which by means of computable methods with the minimum of human intervention will generate from the input text (full text, abstract, or title) a document representative adequate for use in an automatic retrieval system. This is a tall order and can only be partially met. A document will be indexed by a name if one of its *significant* words occurs as a member of that class.

Such a system will usually consist of three parts:

1. removal of high frequency words,
2. suffix stripping,
3. detecting equivalent stems.

The removal of high frequency words, 'stop' words or 'fluff' words is one way of implementing Luhn's upper cut-off. This is normally done by comparing the input text with a 'stop list' of words which are to be removed. The advantages of the process are not only that non-significant words are removed and will therefore not interfere during retrieval, but also that the size of the total document file can be reduced by between 30 and 50 per cent.

The second stage, suffix stripping, is more complicated. A standard approach is to have a complete list of suffixes and to remove the longest possible one.

For example, we may well want UAL removed from FACTUAL but not from EQUAL. To avoid erroneously removing suffixes, context rules are devised so that a suffix will be removed only if the context is right.

'Right' may mean a number of things:

1. the length of remaining stem exceeds a given number; the default is
2. the stem-ending satisfies a certain condition, e.g. does not end with Q.

Many words, which are equivalent in the above sense, map to one morphological form by removing their suffixes. Others, unluckily, though they are equivalent, do not. It is this latter category which requires special treatment. Probably the simplest method of dealing with it is to construct a list of equivalent stem-endings. For two stems to be equivalent they must match except for their endings, which themselves must appear in the list as equivalent.

**For example,**

stems such as ABSORB- and ABSORPT- are conflated because there is an entry in the list defining B and PT as equivalent stem-endings if the preceding characters match.

The assumption (in the context of IR) is that if two words have the same underlying stem then they refer to the same concept and should be indexed as such. This is obviously an oversimplification since words with the same stem, such as NEUTRON AND NEUTRALISE, sometimes need to be distinguished. Even words which are essentially equivalent may mean different things in different contexts. Since there is no cheap way of making these fine distinctions we put up with a certain proportion of errors and assume (correctly) that they will not degrade retrieval effectiveness too much.

It is inevitable that a processing system such as this will produce errors. Fortunately experiments have shown that the error rate tends to be of the order of 5 per cent Surprisingly, this kind of algorithm is not core limited but limited instead by its processing time. The final output from a conflation algorithm is a set of classes, one for each stem detected. A class name is assigned to a document if and only if one of its members occurs as a significant.

A document representative then becomes a list of class names. These are often referred to as the documents *index terms* or *keywords*. Queries are of course treated in the same way. In an experimental situation they can be processed at the same time as the documents. In an operational situation, the text processing system needs to be applied to the query at the time that it is submitted to the retrieval system.

**CONCLUSION**: Thus, we have implemented the Conflation Algorithm.