**Introduction**

KWIC is a Key Word In Context index system. It takes in a list of lines, such as movie titles, and a list of “words to ignore”, generating a Key Word In Context index of the titles. Users can interact with KWIK by providing a text file containing required information or by manually inputting it.

**Design**

KWIC comprises of 10 modules that are designed to process different stages. It follows a batch processing design as each stages will process through all input before proceeding to the next process.

**Limitation & Benefits**

KWIC comprises of multiple modules which process the data and return it to the main control module. This gives KWIC several benefits. First, it supports reuse of modules since each module is capable of performing in isolation provided that it is given input in the form it expects. Second, KWIC is highly extensible. Since the data is always returned back to the main control module, KWIC can easily be extended to fulfill new requirements by inserting new modules to process the data into the main control module. Third, KWIC maintains the intuitive flow of processing as the same data is processed by different modules at different stage of the process similar to a pipeline solution. Fourth, data representation can be changed easily without affecting the rest of the program since each modules is designed to accept an interface rather than a concrete class. Fifth, testing of KWIC can be done easily as each modules are able to be tested individually therefore leading to shorter debugging time.

However, KWIC’s design is not without limitation. KWIC is inefficient in terms of its use of space. In order to better support extension, the data representation of a single line will need to contain at least the original line and the tokenized line, effectively doubling the required space. This in turn translate into longer processing time as modules which are required to process through all the data will need longer time for more data.