**Information Extraction with Genetic Algorithms, *Instruction Manual***

Josh McDonald

**About this Project**

This project was initially designed and programmed with the intent of creating relatively unique software that employs an area known as genetic algorithms to perform tasks related to information extraction. These tasks being, take an article stored in a file and keyword inputs from the user, and use techniques of genetic algorithms to determine what keywords are most relevant to the provided article. As development continued the project expanded to focus more on taking a set of keywords provided by the user, and a set of articles also provided by the user, and determine which keywords / article pairing is the strongest. Before running the software you must have the latest version of java installed: (<http://www.oracle.com/technetwork/java/javase/downloads/index.html> ) then launch the provided jar file either through terminal window or via the batch file included (Windows Users only).

**Running the project / Parameter explanation:**

Upon first running the project, you are presented with a prompt for the name of your article file. This file should be located preferably on your local directory however entering a specific location such as C:\Users\MyName\Documents\Article.txt will work fine as well. The article’s formatting should be standard plaintext, and each article should be designated as such:

 *$name Article Name*

 *Article Content*

 *$end*

The name designator represents an internal variable used to specify one article from the next. *$End* specifies the end of an article, and must follow each article in the file.

Next you will be asked whether or not you would like to enter custom keywords, which is highly likely. Custom keywords can either be typed or loaded from a specific file, keywords.txt, which should be located on your local directory. If you type the keywords into the terminal, then should each be separated by commas, including the last keyword.

Following keyword retrieval the program will ask for the number of keywords to be used per chromosome. The number of keywords you choose must be smaller than or equal to the number of keywords that you input. Keep in mind that a larger number of keywords per chromosome will increase the processing time of the Genetic Algorithm functionality. Next the program will request that you give a number of chromosomes to make up the population. Each chromosome has a set of keywords associated with it that were either randomly generated, or formed from some fusion of it has a floating-point value representing its fitness, which increases with more keyword matches to words in the articles. Afterword’s you will be asked for the population size. The population size is the number of chromosomes to be created. Larger population sizes can yield a variety of results, but take longer to process. Smaller population sizes may find a solution that may not be the best, but will run quickly.

Next you must enter the number of generations to process for. The higher the number entered, the more accurate your results will be, as the GA will have longer time periods to learn which combinations of keywords and articles yield the highest amount of fitness. A good starting value is 40.

Next you will be asked for the mutation and elitism rates. The mutation rate is a floating-point value within the range of 0.0 to 1.0 that is a percentage used during crossover to see if a mutation should be introduced to the system. A mutation in our software is defined as taking an existing chromosome object, and reassigning completely new, randomly chosen keywords. The elitism rate is also a floating-point value between 0.0 and 1.0. The elitism rate can be thought of as the inverse rate of which the population is acceptant of new members of a population. This effectively means, having an elitism rate of around 0.8 means that only the remaining 0.2 or 20% of the previous population will be kept around for the next generation of crossover.

Second to last is the percentage of abstracts that you wish to see. This number spanning from 0 to 100 can be thought of as the percentage differences between the top article’s fitness and the fitness of every other article. A higher number means you will see only the very most relevant articles after the GA has finished processing.

Finally you will be asked if you would like to change the weights used by the fitness function. These weights will affect how the GA treats certain keywords and situations, like having plural case words in an article, but a singular keyword.