Plagiarism is a common problem across all levels of education. Due to the amount of solutions available online, detecting plagiarism in programming assignments can be difficult. This allows students to quickly share solutions with each other. A key strategy for combatting the use of plagiarised solutions is the use of a plagiarism detector.

Plagiarism detection is a concept that has been implemented for common programming languages such as Java, C, C++ etc. Several less common programming languages do not have access to online plagiarism detectors at all. One of these languages is ARM assembly language – a 32-bit reduced instruction set computer instruction set architecture.

This dissertation compares four algorithmic approaches that measuring similarity between files. The results from the experiments conducted shows that using a modified Levenshtein distance approach returns the best results when comparing similarity between ARM assembly language files that have been manually plagiarised. In this experiment Levenshtein returns a recall and precision value of 1 which means it can perfectly differentiate between unique solutions to the same problem and plagiarised files.

Using this method with a correctly calibrated threshold value has created a possibility for an ARM assembly language plagiarism detector to be implemented and integrated into an already existing online resource for plagiarism detection.