Final Year Project

**Software plagiarism detection system using machine learning approach**

**Lian Wang**

**2021**

**Background:**

**In the process of researching about the solution for software plagiarism, legacy software cannot fulfil the need of people. There are two type of software on the market: one analysis the semantics of the source code and compare the code to the resources. Another kind is to record the features of the file, for example the birthmark, and analysis those features. Both are not satisfied. The first one often lack accuracy that need human to examinate the result. The latter requires a high demand of right to access to those features that violate the privacy or refused by the users.**

**Machine learning is new trend on the market. Machine learning is a great tool in analysis the possibilities. The software plagiarism detection is a task about possibility.**

**Literature Review:**

**Moss:**

**Winnowing: Local Algorithms for Document Fingerprinting**

**Question:**

**What is used to be compared with in Moss?**

**Compare within the submissions**

**Compare with the web resource**

**What is the range/ how to choose the web resource used e.g. github?**

**What is hash function?**

**What is its function in the process of similarity check?**

**PIY:**

[Efficient clustering-based source code plagiarism detection using PIY](https://link-springer-com.libproxy.ucl.ac.uk/content/pdf/10.1007/s10115-014-0742-2.pdf)

* **Improved accuracy based on MOSS** 
  + **Tokenization**
  + **Token broken into k-grams**
  + **List of k-gram frequency**
  + **Treat frequency as vectors**
  + **Use vectors to compare similarity**
    - **Manhattan distance**
    - **Cosine distance**
      * **Correspond to the cosine of the angel btw the two vectors representing p(1st doc) and q(2nd doc) in n-dimensional space.**
      * **Divide the dot product of their magnitudes**

**Project goal:**

**By implement machine learning into the software plagiarism detection system, provide a solution that has higher accuracy of detections.**

**Requirements:**

* Must implement a text analysis function to analyse the semantic of the source code
* Must implement an unsupervised machine learning mode that generate an 70% prediction accuracy over training data and 60% over testing data

**Project plan:**

* **Find a database for software plagiarism**
  + [Can’t find a code database](https://ucl-new-primo.hosted.exlibrisgroup.com/primo-explore/fulldisplay?docid=UCL_LG_DSaz%2F33271858&context=L&vid=UCL_VU2&lang=en_US&search_scope=LSCOP_UCL_LG_DS&adaptor=Local%20Search%20Engine&tab=local&query=any,contains,code&offset=0) **in Moodle library**
  + **Does Turnitin has a** [API](https://help.turnitin.com/ithenticate/ithenticate-developer/api/api-guide.htm)**?**
  + **Using students’** [assignment/developers’ work](http://ijmlc.org/papers/50-A243.pdf)
  + **Use a simple program create by myself**
    - **Check ethics on moodle about FYP of github data.**
      * **What can we**
      * **How can use it**
    - **By changing the name of variables, create a plagiarism code for test use**
* **Build a machine learning model**
  + **Collect data**
    - **Where are the dataset in the papers coming from?**
      * [1](https://onlinelibrary-wiley-com.libproxy.ucl.ac.uk/doi/full/10.1002/cpe.5000#cpe5000-bib-0021)
* **Cleanse data**
* **Find a way to represent source code**
* **Choose a model**
* **evaluation**
* **Adjust the parameters**
* **prediction**

**Tasks break down:**

* **Read source file as input**
* **Extract features**
  + **Abstract syntax tree**
  + **Intermediate code generation**
* **Transfer features into matrix**
* **Use logistic regression model to classify source code**

**Weekly report:**

**References:**

**F**