**Similar Items: Plagiarism Detection**

**Initial analysis**

*1. How can we establish plagiarism in python source files? Keep in mind that students often tend to change solely the names of variables and functions within the code.*

* In de eerste plaats tokenizen we de source files. Op die manier worden alle variabelen, klassen- en functienamen vervangen door “ID”-tokens. We zouden deze tokens dan als één karakter kunnen voorstellen (bijv. “$”).
* Eens de broncode uniforme ID’s bevat, kunnen we deze opsplitsen in woorden van een bepaalde lengte die we vervolgens verder in k-shingles samenvoegen.

*2. Explain how Minhash signatures can be computed in a MapReduce environment and argue about the correctness of your approach. Describe in detail how hashes are computed.*

* Indien alle bestanden gelijkmatig als chunks worden verdeeld over de nodes, kan het parallel berekenen van MinHashes voor problemen zorgen. De nodes kunnen vlot de tokenizing-stap en shingling-stap uitvoeren. Bij de chunk-aanpak verliezen we echter de bundeling van een document. Je kan de MinHashing-stap enkel uitvoeren op bit-vectoren die op het volledige document zijn gebaseerd. Om de shingles naar behoren te hashen, zouden we daarom eerst alle chunks moeten defragmenteren in documenten. Deze stap lijkt ons omslachtig.
* Vermits het ontdekken van plagiaat wordt gebruikt binnen de context van schooltaken, lijkt het ons veroorloofd om te veronderstellen dat de documentgrootte niet erg variabel zal zijn. Dit betekent dat we documenten over nodes kunnen verdelen zonder een echt ongelijkmatige werkverdeling te veroorzaken. Alle voorgenoemde stappen kunnen dan op de nodes gebeuren zonder de omslachtige defragmentatiestap. Omdat alle documenten ‘heel’ blijven zullen de MinHash signaturen correct worden berekend.

*3. Present the end-to-end architecture of the pipeline, preferably by means of a visualization. Draw specific attention to the input and output of each of the MapReduce stages involved. Clearly indicate the key used during reduce phases and describe the logic of non-trivial map operations.*