REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : SIOD 12/M2/2019

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **SIOD**

Etude de la similarité des documents électroniques code source pour la détection du plagiat

---

Par :

ZIDI Assia

Soutenu le 7 juillet 2019, devant le jury composé de :

Meadi Med Nadjib	**MCB**	**Président**
Zerarka Med foauzi	**MCA**	**Rapporteur**
Boukhlouf Djemaa	**MCB**	**Examinateur**

الملخص :

يهدف هذا العمل إلى إدراك وتحقيق أداة لدعم اتخاذ القرار، بدء دراسة تشابه الوثائق الإلكترونية للغة البرمجة اللغة سي للسماح باكتشاف السرقة العلمية. تساعد هذه الأداة المدرسين في مهامهم التعليمية بما في ذلك تقييم العمل التطبيقي من حيث التكرار أو الانتحال. تتم ترجمة هذا العمل من خلال دراسة مقارنة تتضمن مقاييس التشابه بين وثيقتين إلكترونيتين في لغة البرمجة سي.

يستخدم النظام مستخدمين اثنين: مسؤول ومدرس. تتم دراسة التشابه في النظام على مرحلتين هما: الدراسة النحوية والدراسة الدلالية.

يتكون من نظامين فرعيين هما: النظام الفرعي "الواحد بواحد" والنظام الفرعي "الواحد بمجموعة". يستخدم النظام الفرعي "الواحد بواحد" دراسة تشابه بناءً على مقارنة وثيقة مصدر ووثيقة مستهدفة. من ناحية أخرى ، يتم ترجمة النظام الفرعي "الواحد بمجموعة" من خلال دراسة التشابه بين وثيقة الإلكترونية في لغة البرمجة مصدر والعديد من شفرة وثائق الإلكترونية في لغة البرمجة . هذه العملية من الدراسة هي سلسلة من الخطوات وهي: التجزئة ، الاستبدال ، الدراسة النحوية والدلالية للمصدر والوثائق المستهدفة. يعتمد تحليلنا على قياسات المسافة ، خاصةً الشعاع وطريقة جب التمام . أما بالنسبة للتحليل الدلالي عن طريق تقييم التسلسل الهيكلي للرموز الوسيطة. نتائج تحليل التشابه هو معدل. هذا المعدل هو معدل السرقة العلمية.

الكلمات المفتاحية: دراسة التشابه، التجزئة، التحليل النحوي، التحليل الدلالي، وثيقة الكترونية في لغة البرمجة، السرقة العلمية، اتخاذ القرار.

Abstract

The present work aims to perceive and realize a tool of decision support starting the study of the similarity of the electronic documents of the type source code in language C to allow the detection of plagiarism. This tool helps teachers in their teaching tasks including the evaluation of practical work in terms of redundancy or plagiarism. This work is translated by comparative study involving measures of the similarity between two electronic documents in language C source code.

The system uses two actors: an administrator and a teacher. The first administers and the second exploits. The study of similarity in the system is done in two phases namely: Syntactic and Semantics.

It is composed of two subsystems namely: a subsystem "One by One" and a subsystem "One by Many". The "One by One" subsystem uses a similarity study based on the comparison of a source job and a target job. On the other hand, the "One by Many" subsystem is translated by studying the similarity between an electronic document source code and several electronic documents target source code. This process of study is a sequence of steps namely: the segmentation, the substitution, the pretreatment for the generation of the syntactic and semantic descriptors of the source and target documents. Our analysis is based on distance measurements, notably that of the vector model and the Cosinus method. As for the semantic analysis by the evaluation of the robustness of the structural sequences of the generated intermediate codes. The results of the similarity analyzes is a rate. This is the rate of plagiarism detected.

Key words: study of the similarity, segmentation, analysis syntactic, analysis semantics, source code, plagiarism, Decision Support.

.

Résumé

Le présent travail a pour but de percevoir et de réaliser un outil d'aide à la décision à partir de l'étude de la similarité des documents électroniques de type code source en langage C pour permettre la détection de plagiat. Cette outil permet d'aider les enseignants dans leurs taches pédagogiques notamment l'évaluation des travaux pratiques en termes de redondance ou plagiat. Ce travail est traduit par étude comparative impliquant des mesures de la similarité entre deux documents électroniques code source en langage C.

Le système utilise deux acteurs: un administrateur et un enseignant. Le premier administre et le second exploite. L'étude de la similarité dans le système est faite en deux phases à savoir: Syntaxique et Sémantique.

Il est composé de deux sous système à savoir: un sous système « One by One » et un sous système « One by Many ». Le sous système « One by One » utilise une étude de la similarité traduite par la comparaison d'un travail source et d'un travail cible. Par contre le sous système « One by Many » est traduit par l'étude de la similarité entre un document électronique code source et plusieurs documents électroniques code source cible. Ce processus d'étude est une séquence d'étapes à savoir: la segmentation, la substitution, le prétraitement pour la génération des descripteurs syntaxique et sémantique des documents source et cible. Notre analyse est basée sur les mesures des distances notamment celle du modèle vectoriel et la méthode du Cosinus. Quant à l'analyse sémantique par l'évaluation de la robustesse des séquences structurelles des codes intermédiaires générés. Les résultats des analyses de la similarité sont un taux. Ce dernier est le taux de plagiat détecté.

Les mots clés: étude de la similarité, segmentation, analyse syntaxique, analyse sémantique, code source, plagiat, aide à la décision.

Remerciements

Avant toute chose, je remercie Dieu, le tout puissant, qui ma donné le courage et la volonté de faire ce travail.

Je tiens à remercier profondément a mon encadreur **Dr.ZERARKA Mohamed Faouzi** pour sa précieuse aide, ses orientations et le temps qu'il m'a accordé pour la direction de mon projet de fin d'études.

Je remercie toutes les personnes qui m'ont aidée de près ou de loin à la réalisation de ce travail.

Dédicaces

Je dédie ce travail,

A mes parents,

Qui ont voulu voir en nous, leurs enfants Toutes les bonnes choses dont ils ont rêvées Et grâce à qui je suis qui je suis.

A mon frère et mes chères sœurs,

Pour leurs encouragements permanents, et leur soutien morals.

A mes copines,

En témoignage de l'amitié qui nous uni et des souvenirs de Tous les moments que nous avons passé ensemble.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Depuis les anciennes années, L'information est désormais perçue comme une ressource indispensable dans tous les secteurs de l'activité humaine, politiques, économiques, administratifs et culturels. Sous la pression des nouvelles technologies, la gestion (création, collecte, stockage, traitement et diffusion) de ces informations a connu et cela depuis des années une véritable révolution. Ces informations sont contenues dans des documents exprimées en plusieurs formats et langages dans des différents supports physiques. Parmi cela les documents scientifiques et techniques qui sont de plus en plus numérisés et deviennent, par conséquent exploitables par des moyens automatisés tel que l'ordinateur.

Dans ce sens, les documents électroniques contiennent plusieurs formats qui peuvent être classés en différentes catégories en fonction de leur mode de création, de leurs descriptions et de leurs contenus. Dans le but de mieux les analysées et plus particulièrement d'étudier leurs similarités pour éviter le problème de la redondance, le problème des droits d'auteurs,. . . .etc. Le document code source est particulier par sa forme, son organisation ses supports électronique, ses modes d'accès et son utilisation. L'usager dit ' le programmeur' peut consulter et mettre a jour(insérer, modifier supprimer) les informations sur ce document d'une façon très économique. Il s'agit d'exprimer les informations du document en un langage évolué permettant ainsi aux professionnels de l'informatique de le comprendre, de le reproduire ou de le modifier aisément.

Depuis l'adoption définitive de la loi pour une République numérique, il est désormais possible de faire une demande de communication de document administratif d'où le code source des logiciels de l'État faisant partie de cette catégorie.

Dans ce contexte, Ce travail se place également dans une optique d'aide à la décision en se basant sur les différentes analyses de la similarité des documents électroniques notamment code source pour répondre aux problèmes de redondance et de plagiat.

Cette analyse de la similarité s'articule sur les méthodes de segmentation des documents utilisant les mesures des différentes distances entre les segments obtenus. Aussi elle dote l'enseignant d'un outils d'aide à l'évaluation des différents travaux pratiques réalises par ces étudiant. Donc, ce travail répond à la problématique que nous procurons et qui fait référence à : 'comment faire une étude de la similarité des documents électronique code source en langage C pour la détection du plagiat ?'

Notre mémoire est composé de cinq chapitres dont le premier consiste à présenter dans sa première parie, l'étude de la similarité avec leurs approches permettant la comparaison des textes, les différents types de la similarité et les Techniques d'analyse de la similarité aussi ce chapitre parle sur le processus d'analyse de la similarité qui est la segmentation, similarité syntaxique et similarité sémantique. La troisième partie consiste à définir un état de l'art sur les méthodes qui applique l'analyse de la similarité.La dernière partie montre les points essentiels

concernant le plagiat.

Le deuxième chapitre présente les concepts de base selon différents points de vue, les fonctions, les opérations, typologie et les types de structure d'un document, un document électronique et un document administratif, ainsi que les concepts de code source et le langage informatique.

Le troisième chapitre s'intéresse notre cas d'étude le document électronique code source en langage C.

Le quatrième chapitre traite la conception de notre système avec les différentes étapes de la détection des documents électronique code source en langage C plagié au sein d'un corpus en termes d'acteurs et de sous systèmes à savoir :

- L'administrateur : c'est le responsable de système qui fait des fonctions principales de préparation des différents support pour une éventuelle étude de la similarité.
- L'enseignant : c'est l'utilisateur principale du système qui va lui permettre l'évaluation d'un travail dirigé par comparaison avec une autre cible du groupe ou avec plusieurs cible du groupe.
- Le sous système « One by One » qui fait la détection de plagiat d'un document code source contre un autre document code cible.
- Le sous système « One by Many » qui fait la détection de plagiat d'un document code source contre un groupe des documents codes cible.

Nous avons réservé le cinquième chapitre à l'implémentation et la présentation de nos expérimentations et résultats des évaluations appliquées à notre système.

enfin , nous terminerons par une conclusion générale et quelques perspectives que nous envisageons entreprendre dans des travaux de futur.

Objectif ciblé

L'objectif de ce travail est de concevoir et implémenter un outil d'aide à la décision à partir de l'étude de la similarité des documents code source pour la détection de plagiat. Autrement dit aider un enseignant en informatique à l'évaluation des travaux pratiques des étudiants remis sous la forme de documents code source dans langage C.

Chapitre 1

Étude de la similarité

1.1 Introduction

Dans ce chapitre, on va présenter dans sa première parie, l'étude de la similarité avec leurs approches permettant la comparaison des textes, les différents types de la similarité et les Techniques d'analyse de la similarité aussi ce chapitre parle sur le processus d'analyse de la similarité qui est la segmentation, similarité syntaxique et similarité sémantique. La deuxième partie consiste à définir un état de l'art sur les méthodes qui applique l'analyse de la similarité.La dernière partie montre les points essentiels concernant le plagiat.

1.2 Similarité

L'analyse de la similarité des Données regroupe aujourd'hui de nombreuses méthodes, et de nombreux outils, qui visent à découvrir l'information « essentielle » contenue dans un texte pour identifier les éléments et trouver leurs similarités et leurs divergences. [38]

1.2.1 Définition de la similarité

La section suivante insiste sur la présentation de quelques définitions accordées à la notion de la similarité.

1.2.1.1 Définition

D'après Larousse la similarité se dit de choses qui peuvent, d'une certaine façon, être assimilées les unes aux autres : Savons, détersifs et produits similaires. [7]

1.2.1.2 Définition

D'après le robert la similarité est à peu prés semblable. [27]

1.2.1.3 Définition

En mathématiques et en informatique, une mesure permettant de comparer des documents code sources, consiste à comparer des chaines de caractères. C'est une métrique qui mesure la similarité ou la dissimilarité entre deux chaînes de caractères. [47]

1.2.1.4 Définition

La similarité entre deux documents électronique code sources est donnée comme étant le pourcentage d'images ou de plans semblables partages par ces documents. Cette mesure est semblable à celle appliquée aux documents textuels, mesure qui consiste a calculer le pourcentage des mots semblables partages. [44]

1.2.2 Définition Mesure de la similarité

En général, c'est une fonction qui quantifie le rapport entre deux objets compares en fonction des points de ressemblance et de différence. Bien entendu, ces deux objets doivent appartenir a une même classe sémantique. [44]

La mesure de la similarité entre deux séquences, considérée comme étant une abstraction au taux de plagiat, doit être robuste aux transformations que peut contenir une version plagiée du code, telles que les permutations et les duplications des segments de code, les insertions et les suppressions des lignes de code, etc. [48]

1.2.3 Définition de Distance

la distance étant une expression du rapport entre deux objets distincts dans le Droit, l'espace et le Temps. Elle le caractérise quantitativement par la mesure à partir d'une unité établie arbitrairement : la seconde, le mètre, le nombre de générations dans une famille... La distance est quantitative. [32]

1.2.4 Techniques d'analyse de la similarité

L'analyse de la similarité contient plusieurs techniques pour aider à détecter la similitude. Donc, nous expliquerons la différence en détail entre les Techniques d'analyse de la similarité. [51]

1.2.4.1 Technique basée sur le texte

Elle prend deux codes sources et compare ses chaînes de caractères une à une. Puis, elle essaie de trouver les séquences qui sont exactement identiques.

1.2.4.2 Technique basée sur l'arbre

Elle permet de convertir le code source sous la forme d'un arbre de syntaxe abstraite (AST) avec un analyseur. L'AST est un arbre qui présente tous les détails (caractéristiques) d'un code source. On utilise l'arbre syntaxique (AST) parce que c'est une manière de représenter le code source. Une fois l'AST créé, on cherche les sous-arbres similaires d'un arbre avec différents algorithmes. Enfin, on retourne le sous-arbre le plus similaire.

1.2.4.3 Technique basée sur le PDG

La technique du PDG (Program Dependency Graph) utilise un graphe de contrôle ou de données. Même si le graphe peut sembler proche d'un arbre, ces deux techniques sont très différentes. L'AST représente un arbre des caractéristiques du code alors que le PDG représente les chemins d'exécution ou le cycle de vie des données.

1.2.4.4 Technique basée sur les métriques

Cette technique compare les métriques du code calculées sur les deux codes à analyser. Une métrique donne une façon de mesurer une caractéristique du code. Par exemple, le nombre de lignes de code, la complexité cyclomatique, etc.

1.2.4.5 Technique de la factorisation

Par exemple, Chilowicz et al. Présentent une technique basée sur les métriques pour détecter la similarité du code source en utilisant la factorisation. On a souvent entendu parler de la factorisation dans le domaine des mathématiques. Pour les mathématiciens, la factorisation consiste à écrire une expression algébrique (notamment une somme), un nombre ou une matrice sous la forme d'un produit. Dans le domaine de l'informatique, on peut utiliser cette méthode pour détecter la similarité des codes sources à réutiliser. En programmation, l'algorithme de la factorisation se base sur le tableau de suffixes. Le facteur n'est pas une expression algébrique ni un chiffre mais sont des fonctions et des sous-fonctions.

1.2.4.6 Approche hybride

Il y a beaucoup d'outils qui utilisent cette approche. Elle consiste à combiner plusieurs techniques pour résoudre un problème.

1.2.5 Comparaison

Plusieurs techniques ont été conçues pour détecter la duplication de code source. La figure suivante compare les différentes techniques de détection des clones pour le code source. [51]

Propriétés	Techniques basées sur le texte	Techniques basées sur l'arbre	Techniques basées sur PDG	Techniques basées sur les métriques	Approche hybride (AST + arbre de suffixes)
Transformation	Supprime les espaces et commentaires	Analyse syntaxique pour créer un AST	Analyseur syntaxique pour créer un PDG	Analyse syntaxique pour créer un AST afin de générer les métriques	Analyse syntaxique pour créer un AST, puis une autre forme pour l'arbre de suffixes
Représentation du code	Filtrées et/ou normalisées	L'AST du programme à partir du texte et de la structure du code	Ensemble de PDG des procédures du système	Ensemble de métriques	AST représenté sous la forme d'un arbre de suffixes
Granularité de la comparaison	Ligne	Nœud de l'arbre	Nœud PDG	Métriques pour chaque méthode/bloc	Jeton de l'AST encodé dans une séquence
Complexité de calcul	Dépend de l'algorithme	Quadratique	Quadratique	Linéaires	Linéaires
Opportunités de réusinage	Bon pour les correspondances exactes	Trouve les clones, bon pour le réusinage	Bon pour réusinage	Inspection manuelle requise	Clones, bon pour le réusinage
Indépendance du langage	Facilement adaptable	Besoin d'un analyseur	Besoin d'un générateur de PDG	Souvent besoin d'un analyseur	Besoin d'un analyseur

FIGURE 1.1 – Techniques d'analyse de la similarité du code [source : Analyse de la similarité du code source pour la réutilisation automatique de tests unitaires à l'aide du CBR université de Laval Québec, Canada

1.2.6 Technique de la similarité basée sur le texte

La similarité dans le code de programmation (textuelle) est un domaine très important pour pouvoir identifier les codes sources similaires dans les conceptions ou les spécifications. [47]

1.2.6.1 Segmentation

Le domaine de la segmentation est un domaine qui a donné lieu à de nombreux travaux ces dernières années. L'idée de segmenter des documents textuels (article de journaux, livres, code source . . .) en blocs de texte est un axe de recherche activement exploré dans les années 90 en Recherche d'Information (RI) textuelle. [41]

1.2.6.1.1 Définition de segmentation Former des groupes homogènes à l'intérieur d'une population. [6]

- Étant donné un ensemble de points, chacun ayant un ensemble d'attributs, et une mesure de la similarité définie sur eux, trouver des groupes tels que :
  - □ Les points à l'intérieur d'un même groupe sont très similaires entre eux.
  - □ Les points appartenant à des groupes différents sont très dissimilaires.
- Le choix de la mesure de la similarité est important.

1.2.6.1.2 objectifs de segmentation La détermination de la structure des codes sources composites en blocs homogènes est un problème très complexe auquel on a souvent tenté de donner des solutions dans des cas bien particuliers. L'organisation des constituants en blocs séparables pour la description du code source constitue la structure physique. La reconnaissance de cette organisation et l'étiquetage des blocs en diverses catégories constituent la phase d'identification de la structure. Elle est connue sous le nom de structuration logique du code source. [36]

1.2.6.1.3 méthodes de la segmentation Il existe deux familles de traitements dédiées à la recherche de telles structures : les méthodes dites « ascendantes » et « descendantes ». Les premières permettent d'extraire des blocs de textes par la fusion de petites composantes (généralement des composantes connexes) jusqu'à obtenir des blocs plus larges. Les secondes permettent de segmenter l'image par des découpages successifs de grandes composantes (blocs de grandes tailles) en composantes plus petites. [36]

méthodes descendantes Ces méthodes sont performantes dans les situations où l'on connaît la structure a priori du document à analyser. C'est la raison pour laquelle, elles s'appliquent essentiellement à des documents très spécifiques et très hiérarchisés, tels les documents administratifs et scientifiques.

méthodes ascendantes La stratégie générale de la segmentation ascendante est fondée sur l'analyse de composantes connexes. Elle consiste à fusionner les morceaux jusqu'à l'assemblage complet de la page du document. La technique la plus répandue est la technique de lissage directionnel ; elle a été proposée par Wong et al.

méthodes hybrides Les méthodes hybrides (mi-ascendantes, mi-descendantes) sont plus efficaces que les méthodes exclusivement ascendantes et descendantes. Dans [Tsujimoto] et dans [Wieser], les auteurs ont mis à contribution cette mixité pour reformer à partir de la fusion d'objets (caractères, symboles) des composantes complètes de lignes et de paragraphes.

1.2.6.2 Similarité syntaxique

Une mesure de la similarité syntaxique permet de comparer des documents textuels en se basant sur les chaines de caractères qui les composent. Par exemple, les chaînes de caractères "voiture" et "voiturier" peuvent être considérées comme très proches, alors que "voiture" et "automobile" pourront être considérées comme très différentes. [47]

1.2.6.2.1 Modèle d'espace vectoriel

La représentation d'un ensemble de documents sous forme de vecteurs dans un espace vectoriel commun est connu sous le nom de modèle d'espace vectoriel (vector space model). dans un modèle d'espace vectoriel, les documents sont représentés comme des vecteurs de caractéristiques représentant les termes qui apparaissent dans la collection. On parle aussi de 'sacs de mots' où les mots sont considérés comme indépendants et où l'ordre est sans importance.
La représentation d'un document sous forme vectorielle se déroule en deux étapes :

- extraire les termes pertinents du document.
- calculer les poids des termes restants.

Extraction des termes pertinents : Il s'agit de pré-traiter le texte des documents textuels en supprimant les mots-vides, la ponctuation et les éventuels 'retours-chariots', de lemmatiser le texte et de le segmenter.

Calcul des poids : La valeur de chaque caractéristique est appelé le poids du terme et est en général une fonction de fréquence de termes dans le document. les termes qui apparaissent le plus fréquemment sont plus importants et donc descriptifs du document. Le poids de chaque terme dans un document peut être obtenu de différentes manières : booléenne, fréquence des termes, tf-idf (Term frequency - Inverse Document Frequency).

- **Méthode booléenne :**De manière booléenne, si un terme existe dans un document alors la valeur qui lui correspond vaut 1, sinon 0. L'approche booléenne est utilisée lorsque chaque terme est d'égale importance et s'emploie uniquement lorsque les documents sont de petites tailles.
- **Fréquence des termes :**Pour la fréquence des termes, le poids d'un terme est obtenu en comptant les occurrences du terme dans le document : tfi ;j représente donc la fréquence du terme i dans le document j.

  □ TF (Term Frequency).

  □ TF-IDF (Term Frequency-Inversed Document Frequency).

  Pour chaque terme t_i et chaque document d_j , la TF(t_i ,d_j) est calculée de différentes manières, par exemple :

  $a-$ En utilisant le nombre total de termes dans le document :

$$TF(t_i, d_j) = \begin{cases} \frac{n_{ij}}{\sum_{k=1}^{k=n}} & \text{si } n_{ij} > 0 \\ 0 & \text{si } n_{ij} = 0 \end{cases}$$

$b-$ En utilisant le maximum des nombres d'occurrences des termes dans le document :

$$TF(t_i, d_j) = \begin{cases} \frac{n_{ij}}{\max(n_{kj})} & \text{si } n_{ij} > 0 \\ 0 & \text{si } n_{ij} = 0 \end{cases}$$

$c-$ En utilisant l'échelle logarithmique pour conditionner le nombre des termes (cette approche est utilisée dans le système Cornell SMART) :

$$TF(t_i, d_j) = \begin{cases} \frac{n_{ij}}{1+\log(1+\log n_{ij})} & \text{si } n_{ij} > 0 \\ 0 & \text{si } n_{ij} = 0 \end{cases}$$

La fréquence inverse de document idf_i du terme t_i est donnée par :
$idf_i = \log(\frac{N}{df_i})$

Le poids TF-IDF d'un terme est donné par :

$W_{ij} = TF_{ij} IDF_j$

1.2.6.2.2 Mesures de la similarité existantes

Une mesure de la similarité est, en général, une fonction qui quantifie le rapport entre deux objets, comparés en fonction de leurs points de ressemblance et de dissemblance. Les deux objets comparés sont, bien entendu, de même type.

Similarité Cosinus : La similarité cosinus est fréquemment utilisée en tant que mesure de ressemblance entre deux documents d1 et d2. Il s'agit de calculer le cosinus de l'angle entre les représentations vectorielles des documents à comparer. La similarité obtenue :

$Sim_{cosinus}(d1, d2) = \frac{\vec{d1}.\vec{d2}}{\|d1\|\|d2\|}$

Coefficient de corrélation de Pearson : Le coefficient de corrélation de Pearson calcule la similarité entre deux documents d1 et d2 comme le cosinus de l'angle entre leurs représentations vectorielles centrées-réduites. La similarité obtenue :

$Sim_{pearson}(d1, d2) = Sim_{cosinus}(\overline{d1} - d1, \overline{d2} - d2)$

où $\overline{d1}$ (resp. $\overline{d2}$) représente la moyenne de d1 (resp. d2).

Distance euclidienne : La distance euclidienne calcule la similarité entre deux documents d1 et d2 comme la distance entre leurs représentations vectorielles ramenées à un seul point.

$Sim_{euclidienne}(d1, d2) = \|d1 - d2\| = \sqrt{\sum_{i=1}^{n}(d1_i - d2_i)^2}$
où n est le nombre total de termes représentés, i.e. la taille des vecteurs.

Coefficient de Jaccard : L'indice de Jaccard ou coefficient de Jaccard [Jaccard, 1901] est le rapport entre la cardinalité (la taille) de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles. Il permet d'évaluer la similarité entre les ensembles. Les documents d1 et d2 sont donc représentés, non pas comme des vecteurs, mais comme des

ensembles de termes. La similarité obtenue :

$Sim_{jaccard}(d1, d2) = \frac{\|d1 \cap d2\|}{\|d1 \cup d2\|}$

Il est aussi possible d'utiliser la représentation vectorielle.

$Sim_{jaccard}(d1, d2) = \frac{\vec{d1}.\vec{d2}}{\|d1\|\|d2\| - d1.d2}$

Distance (d'édition) de Levenshtein : La distance de Levenshtein calcule la similarité entre les représentations sous forme de chaines de caractères des documents d1 et d2. Il s'agit du coût minimal, i.e. du nombre minimal d'opérations d'édition, pour transformer d1 en d2. Les opérations sont les suivantes :

- ☐ substitution d'un caractère de d1 en un caractère de d2,
- ☐ ajout dans d1 d'un caractère de d2,
- ☐ suppression d'un caractère de d1.

Pour obtenir la distance de Levenshtein $Sim_{levenshtein}(d1; d2)$ entre les documents d1 et d2, il s'agit d'associer à chacune de ces opérations un coût. Le coût des opérations est toujours égal à 1, sauf dans le cas d'une substitution de caractères identiques. Notons que cette distance a été étendue pour prendre en compte la grammaire, la phonétique, ...

Indice de Dice : L'indice de Dice mesure la similarité entre deux documents d1 et d2 en se basant sur le nombre de termes communs à d1 et d2.

$Sim_{dice}(d1, d2) = \frac{2N_c}{N1+N2}$

où Nc N_c est le nombre de termes communs à d1 et d2, et N1 (resp. N2) est le nombre de termes de d1 (resp. d2).

1.2.6.2.3 Avantages et Inconvénients

L'approche de la similarité syntaxique contient des avantages et des inconvénients qui sont :

Avantages :

- Les techniques basées sur l'approche syntaxique ne laissent pas de place aux exceptions ;
- Elles sont donc facilement automatisables.

Inconvénients :

- Par définition, les techniques basées sur l'approche syntaxique ne prennent pas en compte la sémantique.
- Les relations syntaxiques sont ignorées. Par exemple, aucune différence n'est faite entre "Pierre aime Maman" et "Maman aime Pierre".
- De même, les rôles sémantiques sont ignorés. Par exemple, dans "La société A achète la société B" et "La société B a été achetée par la société A", seule la forme verbale change.
- Les problèmes liés aux négations (par exemple, "Je suis malade" et "Je ne suis pas malade") et aux antinomies semblent encore difficiles à pallier.

1.2.6.3 Similarité sémantique

La similarité sémantique est un concept selon lequel un ensemble de documents ou de termes se voient attribuer une métrique basée sur la ressemblance de leur signification / contenu sémantique. [47]

1.2.6.3.1 Mesures existantes

Dans cette section, nous dissocions les approches vectorielles, les approches topologiques et les approches statistiques.

Approches vectorielles : Dans cette section, nous dissocions les approches vectorielles, les approches topologiques et les approches statistiques.

- ☐ **Vecteurs sémantiques**

  L'idée consiste à déterminer la sémantique d'un mot en consultant les autres termes utilisés à ses côtés dans des phrases. Une manière simple de le faire est d'utiliser des vecteurs pour représenter le sens des mots, et d'utiliser ensuite des mesures de la similarité vectorielles (comme pour la similarité syntaxique).
  Le plus difficile est d'obtenir de tels vecteurs. Il faut donc construire un ensemble de vecteurs pour chaque mot dans le dictionnaire utilisé.

- ☐ **Bi-clustering**

  La classification double ou co-clustering ou bi-clustering est une technique d'exploration de données non supervisée permettant de segmenter simultanément les lignes et les colonnes d'une matrice. Étant donné un ensemble de r lignes à c colonnes (c'est-à-dire une matrice r _ c), l'algorithme de bi-clustering génère des bi-clusters - un sous-ensemble de lignes qui présentent un comportement similaire sur un sous-ensemble de colonnes, ou vice versa.

Approches topologiques (ou knowledge-based) : Les approches de la similarité de mots basées sur la connaissance s'appuient sur un réseau sémantique de mots, tel que WordNet [Fellbaum, 1998]. Etant donnés deux mots, leur similarité peut être estimée à partir de leurs positions relatives dans la hiérarchie de la base de connaissances. En effet, la structure de la base est un arbre où chaque nœud est un concept, ses enfants sont les hyponymes du concept, et ses parents sont ses hyperonymes. Les concepts peuvent être des noms, des verbes ou des adjectifs.

- ☐ Edge-based.
- ☐ Leacock et Chodorow.
- ☐ Wu et Palmer.
- ☐ Node-based (ou information content-based).
- ☐ Resnik.
- ☐ Lin.
- ☐ Jiang and Conrath.

Approches statistiques (ou corpus-based) : Les mesures basées sur des corpus diffèrent des mesures présentées précédemment car elles ne nécessitent pas la compréhension du vocabulaire ou de la grammaire de la langue d'un texte.
Parmi de telles mesures de la similarité sémantique, nous présentons l'analyse sémantique latente (LSA) [Deerwester et al., 1990] où les co-occurrences de termes dans un corpus sont capturées au moyen d'une réduction de dimension réalisée par une décomposition en valeurs singulières (SVD) sur la matrice termes/documents représentant le corpus ;
l'analyse sémantique explicite (ESA) [Gabrilovich and Markovitch, 2007] qui est une variation du modèle standard vectoriel où les dimensions du vecteur sont directement équivalentes à des concepts abstraits.
D'autres mesures comme la distance normalisée de Google (Normalized Google Distance (NGD)) [Cilibrasi and Vitanyi, 2007] et le no de wikipédia (no of Wikipedia (noW)) [Wong et al., 2006] existent mais ne sont pas présentées en détail.

1.2.6.3.2 Avantages et Inconvénients

Comparativement aux approches syntaxiques, certains inconvénients ont été palliés, d'autres subsistent. En effet, les problèmes liés à la négation et l'antinomie, aux rôles sémantiques inverses et à l'inconsistance logique ne sont toujours pas réglés.
De plus, il est à noter que les approches sémantiques basées sur les corpus ou la connaissance posent des problèmes de stockage et de complexité et sont souvent spécifiques à un domaine donné.

1.3 État de l'art

De nombreux logiciels de recherche de similitudes utilisant plusieurs types de représentations du code et des approches différentes coexistent actuellement. Nous présentons ici une liste des outils disponibles à savoir [34] :

1.3.1 Code source brut

$a-$ **Alignement :**

- □ **DupLoc :**DupLoc permet la visualisation de lignes de code similaires sous la forme de matrice DotPlot où la cellule (i, j) est noircie si les lignes i et j des unités de compilation concaténées sont identiques. Avant l'étape de comparaison ligne par ligne, le code source est légèrement normalisé (lignes vides supprimées).

$b-$ **Autre approche algorithmique :**

- □ **CodeMatch :** CodeMatch utilise directement le code source brut avec une normalisation du formatage : son adaptation à un nouveau langage est rapide. Il réalise des comparaisons par paires d'unité de compilation en calculant des métriques de la similarité basées sur plusieurs critère :
  - Une métrique recherche la plus longue séquence commune d'instructions entre des unités, une instruction étant définie par le premier mot-clé d'une ligne.
  - D'autres métriques cherchent à mettre en évidence des identificateurs partagés (chaînes entières ou sous-séquences) ainsi que des portions de commentaires dupliqués.

1.3.2 Séquences de lexèmes

$a-$ **Metalexèmisation :**

- ☐ **CPD :** CPD (Copy Paste Duplication) est un outil de recherche de clones au sein d'une répertoire de fichiers source. Plusieurs versions ont été implantées utilisant des méthodes différentes de recherche de clones exacts sur séquences de lexèmes :
  - La première version utilisait l'algorithme Greedy String Tiling (sans l'utilisation de fonctions de hachage incrémentales de type Karp-Rabin) afin d'agglomérer des lexèmes consécutifs identiques.
  - La deuxième version utilisait une table de suffixes pour la recherche de sous-chaînes répétées.
  - Quant à la troisième version, considérée comme plus rapide, elle se base sur l'utilisation d'empreintes générée par une fonction de hachage incrémentale.
- ☐ **JPlag :** JPlag est un outil de recherche de similitudes utilisant une représentation des codes source par séquences de lexèmes. Il compare chaque paire d'un jeu de projets soumis en utilisant l'algorithme Running Karp-Rabin Greedy String Tiling [RKR-GST]. Les résultats sont proposés sous la forme de pages HTML avec une sélection des groupes de correspondances les plus longues (les paires de correspondances identiques sont regroupées).
- ☐ **Plaggie :** Plaggie est un logiciel Open Source de recherche de similitudes utilisant l'algorithme de métalexèmisation RKR-GST originel. Les auteurs disent s'être inspiré de JPlag pour le réaliser avec la différence notable de sa disponibilité sous licence GPL.
- ☐ **Moss :** Moss est un service web à sources fermées réalisant une recherche de la similarité sur des séquences de lexèmes. Il utilise une base d 'empreintes pour le jeu de projets étudiés en générant pour chaque k-gram une empreinte ;

$b-$ **Indexation de suffixes :**

- ☐ **Ccfinderx :** CCFinderX est le successeur sous licence libre MIT de CCFinder. Il propose un outil de recherche de clones exacts sur une forme lexèmisée et normalisée du code source en utilisant un arbre de suffixes. Une interface graphique permet la visualisation de correspondances sur le code source avec affichage d'une matrice dotplot globale.

$c-$ **Autre approche algorithmique :**

- ☐ **Unique :** Unique est un logiciel de recherche de chaînes de lexèmes similaires développé en Python et proposé sous licence GPLv3. Une phase de pré-traitement du code source permet d'obtenir une chaîne de lexèmes d'un ensemble de fichiers chargée en mémoire centrale qui sera utilisée pour la recherche. La lexémisation réalisée est basique (découpage en chaînes de caractères nonespace), applicable à tous les langages et sans opération d'abstraction.
- ☐ **SID :** SID est un service web de recherche de la similarité particulièrement destiné à la recherche de cas de plagiat au sein de jeux de projets d'étudiants. Les projets y sont comparés deux par deux afin d'évaluer une métrique de la similarités basée sur la complexité de Kolgomorov.

1.3.3 Arbres de syntaxe

$a-$ **Autre approche algorithmique :**

□ **CloneDigger :** Après lexèmisation du code source, CloneDigger utilise une approche par antiunification afin de grouper des structures de code similaires. Dans un premier temps, toutes les instructions sont insérés dans des groupes, chaque groupe étant représenté par une schéma d'instruction antiunifié.

1.4 Plagiat

Après que nous avons abordé la segmentation textuelle de type code source et l'étude de la similarité sur ces codes segmentés nous avons trouvé en conséquent les codes les plus similaire.
L'analyse de la similarité entre les codes sources segmentés donne par la suite la mesure de taux de la similarité. Ce taux est considéré comme une abstraction au taux de plagiat détecté entre les codes.

1.4.1 Définition de plagiat

La section suivante insiste sur la présentation de quelques définitions accordées à la notion de plagiat qui touche plusieurs domaines.

1.4.1.1 Définition

En France, le CNRTL (Centre National de Ressources Textuelles et Lexicales) définit le plagiat comme « emprunter à un ouvrage (ici ce terme désigne tout produit issu d'un travail) original et donc par métonymie à son auteur, des éléments, des fragments, dont on s'attribue abusivement la paternité en les reproduisant, avec plus ou moins de fidélité, dans une œuvre que l'on présente comme personnelle ». [40]

1.4.1.2 Définition

Le dictionnaire en ligne d'Oxford qualifie le plagiat comme « la pratique de prendre le travail ou les idées d'un autre et de les faire passer comme étant les siennes ». [40]

1.4.2 Définition plagiat textuel

La section suivante met l'accent sur la présentation de quelques définitions relatives à la notion de plagiat dans l'aspect textuel.

1.4.2.1 Définition

Le plagiat textuel, plagiat d'un écrit, est un plagiat impliquant le vol d'une œuvre écrite. Le copier/coller de tout ou partie d'un texte, sans citer sa source. [40]

1.4.2.2 Définition

Recopier mot à mot l'extrait d'un texte sans mettre de guillemets et/ou sans mentionner la source. [8]

1.4.3 Définition plagiat dans les codes source

Cette section présente quelques définitions accordées à la notion de plagiat dans les codes sources dans une classe des étudiants d'informatique.

1.4.3.1 Définition

Parker et Hamblen définissent le plagiat du code source comme étant un reproduction d'un code à partir d'un code existant, avec un nombre restreint de changements. [48]

1.4.3.2 Définition

Plagiat de code source par des étudiants. Habituellement, le document frauduleux est obtenu par copiage d'un document original et par application d'une série de transformation sur ce dernier. [33]

1.4.3.3 Définition

Un code source est dit plagié lorsqu'il est obtenu par application d'une série de transformations sur un code source original. Le code source plagié doit conserver la même fonction que l'original mais peut avoir une forme différente. [33]

1.4.4 Type de plagiat

Il existe de nombreux types de plagiat, mais les formes les plus courantes sont [18] :

1. Le plagiat direct.
2. Payer pour le travail de quelqu'un d'autre.
3. L'auto-plagiat.
4. Paraphraser sans citer la source.
5. Le plagiat « copier-coller ».

1.4.5 Le plagiat dans le milieu académique et l'enseignement

Le copier/coller touche particulièrement les étudiants. En Europe, 34,5% d'entre eux auraient déjà recopié tout ou partie d'un document pour le présenter comme travail personnel. Cette fréquence rejoint celle d'études américano-canadiennes estimant à plus de 36% la proportion d'étudiants de premier cycle et à 24% la proportion d'étudiants du supérieur ayant déjà réutilisé des phrases provenant d'Internet sans en citer la source. Une étude européenne révèle que près d'un étudiant français sur deux (46%) a déjà fait usage du plagiat pendant son cursus, contre environ 33% des étudiants anglais et 10% des étudiants allemands.

Le plagiat n'est pas seulement monnaie courante chez les étudiants, c'est aussi un phénomène existant chez leurs enseignants. À l'instar de l'exemple de la ministre allemande de l'éducation évoqué dans la section précédente, début 2009, la directrice de l'école de journalisme de Sciences Po, Agnès Chauveau, accusée de plagiat elle aussi sur sa thèse, s'est vu remerciée. [40]

1.4.6 La détection automatique de plagiat

On peut définir la détection automatique du plagiat par un système composé de deux tâches successives. La première tâche est la collecte de documents sources candidats (des documents suspects à comparer par la suite) et la seconde est la comparaison (la recherche d'alignements de passages similaires) de documents deux à deux, entre le document suspect en cours d'analyse et chacune des sources renvoyées par la première tâche. [40]

FIGURE 1.2 – Adaptation de la taxonomie de Eissen et Stein (2006) des différents types de plagiat et de leurs moyens de détection.

1.4.7 Mesure du taux de plagiat

La mesure de la similarité entre deux séquences, considérée comme étant une abstraction au taux de plagiat, doit être robuste aux transformations que peut contenir une version plagiée du code, telles que les permutations et les duplications des segments de code, les insertions et les suppressions des lignes de code, etc. [48]

1.5 Conclusion

Ce chapitre a pour but de présenter les différentes approches essentielles pour faire une comparaison entre des documents textuels de type code source segmenté a fin d'obtenir une mesure de la similarité qui peut définir le taux de plagiat entre ces codes sources.
Nous avons présenté, d'une part, la segmentation textuelle. Dans autre part, nous avons étudié la similarité avec leurs approches permettant de comparer des textes. En plus, nous avons donnée un état de l'art. En fin, nous avons parlé sur le plagiat.

Chapitre 2

Généralités sur les documents et le code source

2.1 Introduction

Dans ce premier chapitre, on va survoler sur les points essentiels relatifs à la notion de document et de code source pour orienter à bien notre travail. On va essayer de définir les concepts de base selon différents points de vue, les fonctions, les opérations, la typologie et les types de structure d'un document, un document électronique et un document administratif, ainsi que les concepts de code source et le langage informatique.

2.2 Document

La notion de document se réfère à plusieurs objets ; elle dépend du domaine et du contexte utilisé. Un document est souvent lié aux tous les concepts suivants : information, donnée, fichier, texte, image, papier, article, livre, journal, feuille, page... etc.

2.2.1 Généralités sur les documents

La notion de document peut varier d'un pays à un autre, ou d'un projet à un autre, selon la terminologie qu'utilise chacun d'eux, dans cette section suivante on va fournir des notions générales sur les documents.

2.2.1.1 Définition d'un document

2.2.1.1.1 Définition 1

Avec l'apparition de nouveaux médias, le document est défini comme un "ensemble formé par un support et une information, généralement enregistré de façon permanente, et tel qu'il puisse être lu par l'homme et la machine." (Définition de l'organisation Internationale de Normalisation). Cinq éléments sont indispensables pour définir un document [16] :

- l'objet « document » contient des informations.
- les informations sont structurées de manières lisibles par un homme ou une machine.
- il repose sur un support transportable, reproductible, relativement stable.
- il a une finalité.
- il est fini en termes de contenu.

2.2.1.1.2 Définition 2

Un document est une information consignée qui est produite ou reçue lors d'une activité ou transaction. De plus, afin d'être considéré comme l'évidence de l'activité en question et prouver ainsi son existence, le document doit avoir un contenu, une structure et un contexte donc. [49]

1. **Contenu :** c'est l'information, par exemple : texte, données, symboles, images-sons. etc.

2. **Structure :** c'est l'apparence et l'arrangement du contenu, par exemple : les liens entre les champs, le langage, le style, etc.

3. **Contexte :** c'est l'information de base qui permet de comprendre les environnements techniques et commerciaux entourant le document, par exemple : métadonnées. applications logistiques, provenance.

2.2.1.2 Les éléments d'un document

Le document est certes un porteur d'informations mais la notion de forme est essentielle car elle détermine la présentation de ces informations La définition du terme « document » par l'AFNOR en 2005 est : « objet porteur d'information(s) organisée(s) ». [49]

1. **Contenu :** c'est les informations qui permettent de décrire le sens du document, et définir les relations sémantiques entre les termes, les données, les symboles . . . etc.

2. **Structure :** La structure du document renvoie à son organisation finale, et à sa structuration en différents éléments constitutifs (blocs d'images ou de texte) qui le caractérisent et permettent de l'identifier. La structure du document présente l'information en utilisant différents codes de présentation. Ces codes sont liés à la technologie utilisée (imprimé, numérique, électronique) mais aussi à des choix esthétiques qui font l'originalité du document. [9]

   La structuration d'un document peut être assurée, entre autres moyens, par un découpage en sections et sous-sections, généralement dotées d'un titre, en plus d'assurer la segmentation et l'organisation visuelle du texte, ils contribuent à la construction de son contenu sémantique.
   Aux deux caractéristiques formelles du document postulées ci-dessus correspondent, deux propriétés sémantiques : un document présente un niveau de contenu sémantique et, simultanément, un niveau abstrait de structuration de ce contenu Nous voulons dire par là que le document ne délivre pas son contenu sémantique, mais comme contenu organisé, structuré, hiérarchisé. C'est de cet ensemble que le lecteur construit un discours, c'est-à-dire un modèle mental de ce qui est en train de s'énoncer, au fur et à mesure qu'il lit le document. [45]

3. **Contexte :** Un Contexte documentaire est défini comme une unité textuelle à l'intérieur d'un document (paragraphe, section, chapitre). L'objectif d'un tel contexte est de mieux prendre en compte la structure des documents. Ces Unités textuelles peuvent avoir des relations entre elles, des travaux visent à trouver ces relations et à les représenter par des annotations. [50]

Types de structure d'un document :

Ils existent plusieurs types de structures pour les documents, nous pouvons les regrouper en trois catégories principales. [39]

1. La structure physique qui permet de regrouper les caractéristiques visuelles du contenu. L'ensemble des documents peut être caractérisé selon une typologie, qui propose des critères, des caractéristiques. Connaître ces caractéristiques vous rendra capables de sélectionner les documents les plus pertinents, aux données les plus facilement exploitables, aux contenus les plus appropriés, aux supports les plus facilement consultables.
   La typologie qui suit vous propose d'abord de distinguer les documents selon :
   - la nature de l'information.
   - le support matériel.
   - le mode de consultation.
   - la périodicité.
2. La structure logique décrit l'organisation du contenu sous forme d'éléments logiques où ces éléments sont liés par des relations.
3. La structure sémantique qui permet d'expliciter le sens d'un contenu.

La forme d'un document

Elle constitue la façon dont les informations sont contenues dans un document : lettre, registre, liste, etc. et la notion de normes ou de standards est donc primordiale pour la conservation du contenu. [43]

1. **Norme :**La norme d'un document est définir par ISO semble être une bonne approche pour déterminer la qualité d'un document dans son ensemble et fournir une vue globale satisfaisante. Cependant, la norme ne précise pas de manière explicite comment mesurer les caractéristiques de qualité définies.
2. **Standards :**Un document standard contient des caractéristiques de qualité définir par ISO selon des cadres de référence et de terminologie standardisés qui facilitent la communication concernant la qualité.

Les plus connus, qui nous font nous interroger sur la nature du document par exemple les normes [13] :

1. le code ASCII.
2. les normes et standards de réseau et de protocoles de télécommunications (ETHERNET, ATM, TCP-IP, HTTP...).
3. les normes de balisage des textes et documents (SGML, HTML, XML, TEI, HyTime...).
4. les normes et standards spécifiant les supports et formats logiques et matériels de l'information (disquettes, CD-Rom, DVD, ZIP...).
5. les standards de traitement de textes, etc...

2.2.2 Fonction d'un document

Le document a deux fonctions principales. [17]

1. **La conservation de l'information :** le document est une trace de l'activité du travail d'une personne ou d'un organisme. Il constitue, de manière provisoire ou définitive, un référent qui supplée la mémoire de l'homme. Dans certains cas, il joue le rôle de preuve.
2. **La communication de l'information :** le document contribue à la diffusion du savoir humain, à l'apprentissage et à la découverte. Il fait connaître le point de vue d'un auteur à partir duquel émergeront d'autres points de vue.

2.2.3 Opération sur le document

Les opérations de base d'un tel document sont [49] :

- **l'acquisition et l'indexation :** Le but de l'indexation est de pouvoir classer les documents dans des armoires, tables similaires ou dans les bases de données pour les retrouver rapidement.
- **L'entreposage :** L'entreposage des documents est une façon plus économique et sécuritaire d'entreposer vos dossiers et documents irremplaçables. Avec toute option d'entreposage de documents, d'importants facteurs sont à considérer :
  - ☐ L'accessibilité.
  - ☐ La protection de la confidentialité.
  - ☐ La gestion de la rétention.
- **la recherche :** Un document n'a aucune valeur si on ne peut pas le retrouver quand on en a besoin. Toute organisation ou tout individu qui souhaite gérer des documents doit imaginer un plan de classement qui permettra de retrouver le ou les documents recherchés. Il ya : La recherche directe, La recherche par mots-clés , Les recherches en texte intégral, La recherche multicritère , La recherche à l'aide d'un plan de classement.
- **la consultation :** La consultation s'effectue de plusieurs façons afin de répondre aux différents besoins des utilisateurs . Deux aspects sont les plus utilisés : L'affichage et La navigation au sein des documents.
- **la modification :** Les modifications se font à l'aide d'éditeurs spécialisés. Ces outils peuvent être ceux qui ont servi à la création des documents.
- **la restitution et la duplication :** Le système doit aussi permettre la multi copie.

2.2.4 Typologie d'un document

Une image de document est composée de déférentes entités physiques ou régions telles que des blocs de texte, des lignes, des mots, des chiffres, des tableaux ainsi qu'un fond. Nous pouvons également assigner des étiquettes fonctionnelles ou logiques telles que des phrases, des titres, des légendes, des noms d'auteurs et des adresses à certaines de ces régions.
Le processus d'extraction de la structure logique et de la structure physique de documents consiste à décomposer une image de document en régions et à comprendre leur fonction et leurs relations dans le document. Le traitement est effectué en plusieurs étapes : le prétraitement, la segmentation de la page (extraction de la structure physique), l'étiquetage logique des segments (extraction de la structure logique). [46]

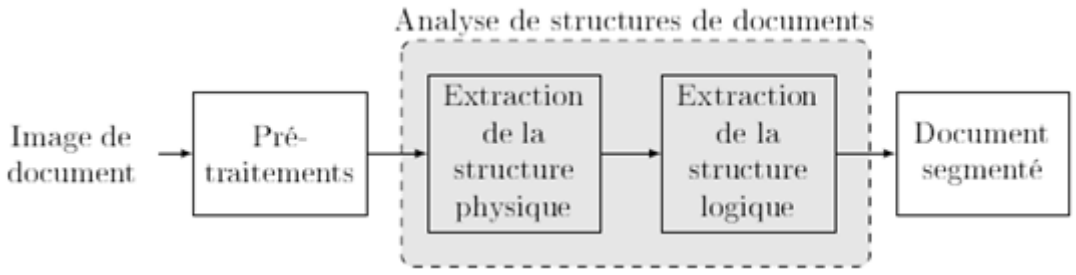


FIGURE 2.1 – Typologie d'un document.

2.3 Document électronique

Un document électronique est tout à fait différent d'un document traditionnel sur papier. Mais bien qu'ayant leurs propres caractéristiques qui les différencient des documents traditionnels.

2.3.1 Généralités sur les documents électroniques

La notion de document électronique est une notion très vaste et complexe, c'est pourquoi on trouve plusieurs définitions et des éléments de base. Dans cette section suivante on va fournir des définitions et des notions générales sur les documents électroniques.

2.3.1.1 Définition d'un document électronique

La section suivante fournis quelques définitions sur la notion de document électronique.

2.3.1.1.1 Définition 1

Un document électronique est un document qui peut être manipulé, transmis ou traité par un ordinateur numérique .Ainsi, le qualificatif «électronique» fait référence au mode de représentation des informations, c'est le mode numérique par opposition au mode analogique. [49]
Les origines des documents électroniques sont diverses :

- documents non numériques (papier, vidéo, son, etc.) qui sont numérisés par captage de L'image.
- documents numériques issus des traitements de texte, tableurs, logiciels de CAO, etc.
- documents créés à partir de différents supports numériques et non numériques (palette graphique permettant en même temps le dessin électronique et la numérisation des dessins, etc.).

2.3.1.1.2 Définition 2

L'ISO (International Standard Organization) définit le document comme étant "l'ensemble d'un support d'information et de données enregistrées sur celui-ci sous une forme en général permanente et lisible par l'homme ou par une machine". Elle définit le document électronique comme étant un "document existant sous forme électronique de manière à être accessible par des installations de traitement de données". [49]

2.3.1.2 Les éléments d'un document électronique

Un document électronique est tout à fait différent d'un document traditionnel sur papier. Mais bien qu'ayant leurs propres caractéristiques qui les différencient des documents traditionnels. Les caractéristiques qui distinguent les documents électroniques des documents traditionnels, sont les suivantes :

1. **Contenu :**le contenu d'un document électronique est représenté par un code binaire qui doit être transformé afin que le document puisse être lisible par les humains. le contenu d'un document électronique est consigné sur un support ou même plusieurs (comme c'est le cas du multimédia : texte+son+vision) et peut être transféré d'un support à un autre souvent d'un type différent. [49]

2. **Structure :**La structure du document désigne le pôle formel lié à l'organisation finale du document et à sa structuration en différents éléments constitutifs (blocs d'images ou de texte) qui le caractérisent et permettent de l'identifier. la structure matérielle d'un document électronique ne l'est pas et elle dépend du choix de l'auteur du document. du système informatique et de l'espace utilisable sur le dispositif de stockage. C'est pourquoi il faudrait construire une structure logique du document permettant l'identification du document et la représentation de chacun de ses éléments constitutifs. [39]

3. **Contexte :**Un contexte d'un document électronique est défini comme un acte textuel, c'est à dire tout ce qui concerne l'intérieur d un document a pour but de mieux prendre en considération la structure des documents électroniques.

Types de structure d'un document électronique

Ils existent plusieurs types de structures pour les documents électroniques, nous pouvons les regrouper en des catégories principales qui sont :

1. **La structure physique/visuelle :**On appelle structuration physique ou visuelle d'un contenu une inscription décrivant la façon dont ce contenu doit être présenté sur un support donné. [23] Le processus d'extraction de la structure physique décompose une image de document en une hiérarchie de régions homogènes pour laquelle chaque région est segmentée de façon itérative en sous-régions d'autres types spécifiques.

2. **La structure logique :**On appelle structuration logique d'un contenu une inscription explicitant la structure de ce contenu en fonction de son organisation et des attributs intrinsèques qui le caractérisent et non en fonction de propriétés de présentation sur un support dans le but d'effectuer une segmentation et une compréhension unique du document. [25]

3. **La structure sémantique :**La structure sémantique exprime l'organisation du contenu sémantique du document. Donc la structure sémantique est l'organisation du contenu sémantique du document textuel. [42]

La forme d'un document électronique

La forme d'un document électronique est comme la forme d'un document traditionnel consiste à conserver les informations et le contenu. Elle a des normes et standards définir par ISO pour déterminer la qualité et les caractéristiques de cette qualité.

2.3.2 Différents Formats de documents électronique

Les formats de fichier ouverts et propriétaires sont bien nombreux à cause de la variété des périphériques utilisés pour la présentation des documents. Document Editor prend en charge les formats les plus populaires. [10]

- **DOC :** L'extension de nom de fichier pour les documents du traitement textuel créé avec Microsoft Word.

- **DOCX (Office Open XML) :**Le format de fichier compressé basé sur XML développé par Microsoft pour représenter des feuilles de calcul et les graphiques, les présentations et les documents du traitement textuel.

- **ODT :**Le format de fichier du traitement textuel d'Open Document, le standard ouvert pour les documents électroniques.

- **RTF (Rich Text Format) :** Le format de fichier du document développé par Microsoft pour la multiplateforme d'échange des documents.

- **TXT :**L'extension de nom de fichier pour les fichiers de texte contenant habituellement une mise en forme minimale.
- **PDF (Portable Document Format) :**Format de fichier utilisé pour représenter les documents d'une manière indépendante du logiciel, du matériel et des systèmes d'exploitation.
- **HTML (HyperText Markup Language) :**Le principal langage de balisage pour les pages web.
- **EPUB (Electronic Publication) :**Le format ebook standardisé, gratuit et ouvert créé par l'International Digital Publishing Forum.
- **XPS (Open XML Paper Specification) :**Le format ouvert de la mise en page fixe, libre de redevance créé par Microsoft.
- **DjVu :**Le format de fichier conçu principalement pour stocker les documents numérisés, en particulier ceux qui contiennent une combinaison du texte, des dessins au trait et des photographies.

2.3.3 Fonction d'un document électronique

Le document électronique a trois fonctions principales [49] :

1. **reproductibles :**un même document peut être reproduit sur plusieurs écrans ou imprimantes simultanément, dupliqué sur support magnétique, transféré sur une autre machine, etc ;
2. **modifiables :**on peut faire du couper-coller, des remises en page, etc ;
3. **transmissibles :** par les réseaux informatiques locaux ou à grande distance.

2.3.4 Opération sur le document électronique

Les opérations de base d'un tel document électronique sont [49] :

- Indexation.
- Support.
- Stockage.
- Diffusion et consultation.
- Archivage et suppression.
- Sécurité.

2.4 Document administratif

Un document administratif est l'un des types de document qui peut être un document traditionnel ou un document électronique. Le document administratif apparait comme le support privilégié de l'action de l'administration.

2.4.1 Généralités sur les documents administratifs

La notion de document administratif est très largement entendue à revoir, et touche les documents produits ou reçus par l'administration. Le document administratif peut être un dossier, rapport, étude, mémoire . . . etc. Dans cette section on va fournir une généralisation sur les documents administratifs.

2.4.1.1 Définition d'un document administratif

Dans cette section nous présentons quelques définitions et notions de document administratif.

2.4.1.1.1 Définition 1

Un document administratif doit être élaboré ou détenu [21] :

- par une administration (État, collectivité territoriale, établissement public),
- par un organisme privé gérant un service public (les caisses de Sécurité sociale, Pôle emploi, un office public de HLM, etc.)
- dans le cadre de missions de service public.

Par exemple :

- dossiers, rapports et études,
- comptes rendus et procès verbaux,
- statistiques,
- directives, instructions et circulaires,
- notes et réponses ministérielles,
- avis et décisions, etc.

2.4.1.1.2 Définition 2

Il s'agit de tous les documents produits ou reçus par l'administration, leur forme étant indifférente. Ils peuvent donc prendre la forme d'un écrit (dossiers, rapports, études, comptes rendus, procès-verbaux, statistiques, directives, instructions, circulaires...) mais également celle d'un enregistrement sonore ou visuel ou d'un fichier numérique ou informatique. Sont également concernées les informations contenues dans des fichiers informatiques et qui peuvent en être extraites par un traitement automatisé d'usage courant. [28]

2.4.1.1.3 Définition 3

Pour la Commission d'accès aux documents administratifs (CADA), « le document administratif informatisé recouvre les documents existant sur support informatique ou pouvant être obtenus par un traitement automatisé d'usage courant.» [20]

2.4.1.2 Les éléments d'un document administratif

Le document administratif apparait comme le support privilégié de l'action de l'administration. Chaque document a un objet bien déterminé qui en motive la rédaction. le document administratif doit avoir un contenu, un contexte et une structure.
La structure de chaque document obéit à des règles de présentation. Certaines parties de la structure sont obligatoires. Il existe un certain nombre de caractères communs à l'ensemble des documents administratifs. D'autres caractères sont spécifiques à certains documents.

Types de structure d'un document

Le document administratif contient aussi une structure physique/visuelle, La structure logique et La structure sémantique.

2.4.2 L'accès à un document administratif

L'article 15 de la Déclaration des droits de l'homme et du citoyen de 1789 prévoit que « la société a le droit de demander compte à tout agent public de son administration ».A ce titre, la loi du 17 juillet 1978 intitulée « de la liberté d'accès aux documents administratifs ». [29]

- **Étape 1 :**Demande de communication à l'administration ou à l'organisme privé chargé d'une mission de service public.
- **Étape 2 :**La Saisine de la CADA.
- **Étape 3 :**Recours pour excès de pouvoir devant le juge administratif.

2.4.3 Document communicable et document non communicable

En vertu de la loi du 17 juillet 1978, de la république française d'accès aux documents administratifs est la règle et le secret l'exception. Les règles de communication ont été définies par la loi, qui a garanti aux citoyens un droit d'accès très large aux documents détenus par les administrations. [14]

2.4.3.1 Document communicable

En présence d'un document administratif, il convient ensuite de déterminer s'il est effectivement communicable. Quatre grands principes s'appliquent en la matière.

1. Document achevé (dossier, rapport, étude, compte-rendu, etc.).
2. Document préparatoire à une décision.
3. Archive publique.
4. Document concernant une personne nommément désignée.

2.4.3.2 Document non communicable

Document comportant des mentions sensibles doc les documents administratifs suivants ne sont pas communicables [29] :

1. Avis du Conseil d'État et des juridictions administratives.
2. Document d'une juridiction financière (Cour des comptes, chambre régionale des comptes).

3. Document d'instruction du Défenseur des droits.

4. Document dont la consultation ou la diffusion porterait atteinte au secret des délibérations du gouvernement, de la défense nationale, à la conduite de la politique extérieure, à la sûreté de l'État, à la sécurité publique, à la monnaie et au crédit public, à la recherche et à la prévention d'infractions.

2.4.4 Avantages d'un document administratif

La conservation des divers documents administratifs se fait en fonction de leur durée de validité et du temps imparti pour une action en justice pour objectif de [37] :

- Augmenter l'efficacité administrative et la productivité.
- Identifier les documents de l'organisme.
- Identifier les documents essentiels.
- Identifier les documents contenant des informations nominatives.
- Identifier les documents à valeur historique ou de recherche pour lesquels une conservation permanente est requise.
- Indiquer la durée de conservation des documents à la phase active, semi-active et leur disposition lorsqu'ils sont devenus inactifs.

2.5 code source

Depuis l'adoption définitive de la loi pour une République numérique, il est désormais possible de faire une demande de communication de document administratif. Le code source des logiciels de l'État tombe dans cette catégorie.

2.5.1 Définition d'un code source

Le code source est un intermédiaire entre l'homme et la machine. Il permet de faire effectuer des taches à une machine programmable en utilisant des concepts proches de la pensée humaine.La section suivante fournis quelques définitions sur la notion de code source.

2.5.1.1 Définition 1

Le code source est un texte qui détaille les instructions d'un programme informatique dans un langage de programmation compréhensible et utilisable par l'homme. Il traduit les instructions qu'a souhaité laisser le programmeur lors de la conception du programme. [5]

2.5.1.2 Définition 2

Le code source est l'objet de la programmation informatique et peut être défini comme un ensemble de commandes informatiques humainement lisibles« écrites » dans un langage de programmation de haut niveau (Krysia et Grzesiek, 2008). [35]

2.5.1.3 Définition 3

Le code source est le fichier qui a permis au développeur de programmer le logiciel, grâce à des lignes écrites en anglais dans un langage particulier, qui sera compris et ensuite compilé en un programme. [15]

2.5.1.4 Définition 4

Ensemble d'instructions à la source d'un programme informatique. Le code source peut être soit compilé (transformé en un fichier exécutable directement par un système d'exploitation), soit interprété (exécuté instruction après instruction par un logiciel spécifique). Du choix du langage de programmation dépend la manière dont le code source est traduit. [4]

2.5.2 langages informatiques

2.5.2.1 Brève histoire des langages de programmation

Bref rappel de l'apparition des langages les plus marquants car il y a énormément de langages [3] :

Année	Langage informatique	Développeur
Avant 1950	-les chinois savaient calculer, automatiser (à la main) des calculs-le principe des itérations successives dans l'exécution d'une opération.-Fortran, Cobol et Lisp.	-El Khawarizmi.-Ada Lovelace.
Les années 1950	- En 1950, l'invention de l'assembleur.- Fortran ,Lisp, Cobol, [Algol].	- Maurice V. Wilkes de l'université de Cambridge.
Les années 1960	- En 1962 :Apl. - En 1964 : Basic. - :PL/1.	- Kenneth Iverson. - Thomas Kurtz et John Kemeny.
Les années 1970	- Depuis 1968 : PASCAL.- En 1978 : langage C.	- Niklaus WIRTH - Kernighan et Ritchie
Les années 1980	- En 1983 : le langage C++.-En 1986 : Eiffel. - En 1988 : les langages Tcl/Tk. - En 1989 : HTML (Hypertext Markup Language).	- Bjarn Stroustrup.- Bertrand Meyer.- John Osterout.- Tim Berners-Lee.
Les années 1990	- En 1995 : Javascript, le langage Php.- Mysql et PosgresSql.	- Brendan Eich - Rasmus Lerdorf
Les années 2000	- C#. - Java,Delphi, Php, Perl.	-Bill Gates.
Les années 2010	- Matlab, Scilab, R, Ruby, Python et librairies Javascript .	- Cleve Moler - Guido van Rossum

TABLE 2.1 – l'apparition des langages informatiques

2.5.2.2 Définition d'un langage informatique

Les langages de programmation permettent de décrire d'une part les structures des données qui seront manipulées par l'appareil informatique, et d'autre part d'indiquer comment sont effectuées les manipulations, selon quels algorithmes. Ils servent de moyens de communication par lesquels le programmeur communique avec l'ordinateur.

2.5.2.2.1 Définition 1

On appelle « langage informatique » un langage destiné à décrire l'ensemble des actions consécutives qu'un ordinateur doit exécuter. Un langage informatique est ainsi une façon pratique pour nous (humains) de donner des instructions à un ordinateur. [1]

2.5.2.2.2 Définition 2

C'est un langage qui sert à décrire les actions qu'un ordinateur doit réaliser. Ces actions sont innombrables et variées. Il peut s'agir aussi bien d'ouvrir une fenêtre avec la souris, d'effacer un mot dans un texte, de tirer sur un adversaire dans un jeu ou de modifier la définition de l'écran. [22]

2.5.2.3 Langages impératifs et fonctionnels

On distingue habituellement deux grandes familles de langages de programmation, selon la manière de laquelle les instructions sont traitées [1] :

1. les langages impératifs.
2. les langages fonctionnels.

2.5.2.3.1 Langage impératif

Un langage impératif organise le programme sous forme d'une série d'instructions, regroupées par blocs et comprenant des sauts conditionnels permettant de revenir à un bloc d'instructions si la condition est réalisée.

Programmation structurée (ou procédurale) : La programmation structurée est possible dans n'importe quel langage de programmation procédural, mais certains, comme le Fortran IV, s'y prêtaient très mal. Vers 1970, la programmation structurée devint une technique populaire, et les langages de programmation procéduraux intégrèrent des mécanismes rendant aisée la programmation structurée. Parmi les langages de programmation les plus structurants, on trouve PL/I, Pascalet, plus tardivement, Ada.

Programmation orientée objet : La programmation orientée objet (POO) ou programmation par objet, est un paradigme de programmation informatique qui consiste en la définition et l'assemblage de «briques logicielles» appelées objets. Un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre.

2.5.2.3.2 Langage fonctionnel

Un langage fonctionnel (parfois appelé langage procédural) est un langage dans lequel le programme est construit par fonctions, retournant un nouvel état en sortie et prenant en entrée la sortie d'autres fonctions. Lorsque la fonction s'appelle elle-même, on parle alors de récursivité.

2.5.2.4 Interprétation et compilation

Les langages informatiques peuvent grossièrement se classer en deux catégories [30] :

1. les langages interprétés
2. les langages compilés.

2.5.2.4.1 Langage interprété

Un langage informatique est par définition différent du langage machine. Un programme écrit dans un langage interprété a besoin d'un programme auxiliaire (l'interpréteur) pour traduire au fur et à mesure les instructions du programme.

2.5.2.4.2 Langage compilé

Un programme écrit dans un langage dit « compilé » va être traduit une fois pour toutes par un programme annexe, appelé compilateur, afin de générer un nouveau fichier qui sera autonome, c'est-à-dire qui n'aura plus besoin d'un programme autre que lui pour s'exécuter ; on dit d'ailleurs que ce fichier est exécutable.

2.6 Conclusion

Dans ce chapitre nous avons présenté les notions de base sur le document, langage informatique et le code source, nous avons expliqué dans un premier temps les concepts de base, les fonctions, les opérations, typologie et les types de structure d'un document, un document électronique et un document administratif. Aussi, nous avons parlé du langage informatique et le code source en générale.

Chapitre 3

cas d'étude : document code source en langage C

3.1 Introduction

Le langage C a été créé au milieu des années 1970, aux Bell Laboratories, par Brian Kernighan et Denis Ritchie, initialement pour écrire le système d'exploitation Unix. La première version a été rendue disponible vers 1977, la référence syntaxique étant alors l'ouvrage informel publié par Kernighan et Ritchie, The C Programming Language.
La syntaxe a évolué dans les années 1980, donnant lieu à une multitude de variations,extensions constructeurs, spécificités machines, etc, désigne la syntaxe officialisée par la norme X3.159-1989 de février 1990. [26]

3.2 Elément de définition administratif

3.2.1 Définition d'un langage C

Le langage C est un langage impératif classique de la famille d'Algol. Le typage est statique. La gestion de la mémoire est manuelle. Il est très riche en opérateurs. Il permet des abstractions sous forme de fonctions. Le langage est conçu pour la programmation système, et donc pour que le programmeur maîtrise totalement la disposition des objets en mémoire. C'est par contre un inconvénient pour la programmation d'applications. Grâce à des opérations d'un niveau relativement bas, le code généré par la plupart des compilateurs est assez rapide. [26]

3.2.2 Fichier Source et exécutable

Un fichier peut être défini comme une entité regroupant un ensemble d'informations, stockée sur un support physique (disque par exemple)et manipulable grâce a un système d'exploitation. On peut distinguer différents type de fichiers. [26]

- Des fichiers exécutables (applications).
- Des fichiers binaires : fichiers objets.
- Des fichiers textes (ASCII).

L'objectif d'un programmeur est bien sur d'arriver générer (puis exécuter) un fichier exécutable. Ceci passe par plusieurs étapes. que nous allons décrire dans le cas d'un programme en langage C.

- La première étape consiste à écrire le programme(on parle de source) dans un fichier texte à l'aide d'un éditeur .En C, on donne l'extension .c .
- La deuxième étape est l'étape de précompilation. Elle consiste à traiter les directives de compilation (comme l'inclusion de fichiers d'entête de bibliothèques).
- étape suivante est la compilation. Elle consiste à transformer les instructions du programme en langage compréhensible par le processeur (langage machine). Elle génère un fichier binaire dit fichier objet.
- La dernière étape consiste à exécuter l'édition de liens .Le code généré à la compilation est complété par le code des fonctions des bibliothèques utilisées .C'est seulement après cette étape que l'on génère un fichier exécutable.

3.3 Structure du langage

La structure du langage C est comme suit [26] :

1. **les identificateurs :** Le rôle d'un identificateur est de donner un nom à une entité du programme. Plus précisément, un identificateur peut désigner :
   - ☐ un nom de variable ou de fonction,
   - ☐ un type défini par typedef, struct, union ou enum,
   - ☐ une étiquette.

   Un identificateur est une suite de caractères parmi :
   - ☐ les lettres (minuscules ou majuscules, mais non accentuées),
   - ☐ les chiffres,
   - ☐ le blanc souligné .
2. **les mots-clefs :** Un certain nombre de mots, appelés mots-clefs, sont réservés pour le langage lui-même et ne peuvent pas être utilisés comme identificateurs. L'ANSI C compte 32 mots clefs :

auto	**const**	**double**	**float**	**int**	**short**	**struct**	**unsigned**
break	**continue**	**else**	**for**	**long**	**signed**	**switch**	**void**
case	**default**	**enum**	**goto**	**register**	**sizeof**	**typedef**	**volatile**
char	**do**	**extern**	**if**	**return**	**static**	**union**	**while**

FIGURE 3.1 – les mots-clefs dans langage C

3. **les constantes :** Une constante est une valeur qui apparaît littéralement dans le code source d'un programme, le type de la constante étant déterminé par la façon dont la constante est écrite. Les constantes peuvent être de 4 types : entier, flottant (nombre réel), caractère, énumération. Ces constantes vont être utilisées, par exemple, pour initialiser une variable.

4. **les chaînes de caractères :** Une chaîne de caractères est une suite de caractères entourés par des guillemets.Une chaîne de caractères peut contenir des caractères non imprimables, désignés par les représentations vues précédemment.

5. **les opérateurs :** Deux types d'opérateurs peuvent être appliqués aux données(ou opérandes) :

   - ☐ Les Opérateurs unaires.
   - ☐ Les opérateurs binaires.

   A) **Les opérateurs arithmétiques :** Les opérateurs arithmétiques classiques sont l'opérateur unaire (changement de signe) ainsi que les opérateurs binaires.

Opérateur arithmétique	Description
+	addition
-	soustraction
*	multiplication
/	Division
%	reste de la division (modulo)

TABLE 3.1 – Les opérateurs arithmétiques en langage C

   B) **Les opérateurs relationnels :** Les opérateurs relationnels en langage C sont :

Opérateur relationnel	Description
>	strictement supérieur
>=	supérieur ou égal
<	strictement inférieur
<=	inférieur ou égal
==	égal
!=	différent

TABLE 3.2 – Les opérateurs relationnels en langage C

   C) **Les opérateurs logiques booléens :**Les opérateurs logiques booléens en langage C sont :

opérateur logique booléen	Description
&&	et logique
||	ou logique
!	négation logique

TABLE 3.3 – Les opérateurs logiques booléens en langage C

   D) **- Les opérateurs d'affectation composée :** Les opérateurs d'affectation composée sont :

opérateur d'affectation composée	Description
+=	affectation additionneur
-=	affectation soustracteur
*=	affectation multiplicateur
/=	affectation diviseur
%=	affectation modulo
«=	affectation décalage à gauche
»=	affectation décalage à droite

TABLE 3.4 – Les opérateurs d'affectation composée en langage C

E) **Les opérateurs d'incrémentation et de décrémentation :** Les opérateurs d'incrémentation ++ et de décrémentation – s'utilisent aussi bien en suffixe (i++) qu'en préfixe (++i). Dans les deux cas la variable i sera incrémentée, toutefois dans la notation suffixe la valeur retournée sera l'ancienne valeur de i alors que dans la notation préfixe se sera la nouvelle.

F) **L'opérateur virgule :** Une expression peut être constituée d'une suite d'expressions séparées par des virgules : expression1, expression2, ... , expressionN.

3.4 Structure source

Un programme C se présente de la façon suivante :

```
[directives au préprocesseur]

[déclarations de variables externes]

[fonctions secondaires]

main()
{
    déclarations de variables internes
    instructions
}
```

FIGURE 3.2 – Structure code source langage C

3.5 Commentaire

Un commentaire débute par /* et se termine par */. Par exemple : /* Ceci est un commentaire */
ou débute par // . Par exemple : //Ceci est un commentaire

On ne peut pas imbriquer des commentaires. Quand on met en commentaire un morceau de programme, il faut donc veiller à ce que celui-ci ne contienne pas de commentaire.

3.6 Procédures et fonctions

Un sous-algorithme est un bloc faisant partie d'un algorithme. Il est déclaré dans la partie entête (avant le début de l'algorithme) puis appelé dans le corps de l'algorithme.Un sous-algorithme peut se présenter sous forme de fonction ou de procédure. [11]

3.6.1 Procédure

Une procédure est un bloc d'instructions nommé et déclaré dans l'entête de l'algorithme et appelé dans son corps à chaque fois que le programmeur en a besoin.

3.6.2 fonction

Une fonction est un bloc d'instructions qui retourne obligatoirement une et une seule valeur résultat à l'algorithme appelant. Une fonction n'affiche jamais la réponse à l'écran car elle la renvoie simplement à l'algorithme appelant.

3.7 Les fonctions d'entrées-sorties classiques

Il s'agit des fonctions de la librairie standard stdio.h utilisées avec les unités classiques d'entrées-sorties, qui sont respectivement le clavier et l'écran. Sur certains compilateurs, l'appel à la librairie stdio.h par la directive au préprocesseur #include <stdio.h>. [26]

3.7.1 La fonction d'écriture « printf »

La fonction printf est une fonction d'impression formatée, ce qui signifie que les données sont converties selon le format particulier choisi. Sa syntaxe est :
printf("chaîne de contrôle",expression1,...,expressionN) ;
La chaîne de contrôle contient le texte à afficher et les spécifications de format correspondant à chaque expression de la liste. Les formats d'impression en C sont donnés dans la table suivante :

Format	conversion en	écriture
%d	int	décimale signée
%ld	long int	décimale signée
%u	unsigned int	décimale non signée
%lu	unsigned long int	décimale non signée
%o	unsigned int	octale non signée
%lo	unsigned long int	octale non signée
%x	unsigned int	hexadécimale non signée
%lx	unsigned long int	hexadécimale non signée
%f	double	décimale virgule fixe
%lf	long double	décimale virgule fixe
%e	double	décimale notation exponentielle
%le	long double	décimale notation exponentielle
%g	double	décimale, représentation la plus courte parmi %f et %e
%lg	long double	décimale, représentation la plus courte parmi %lf et %le
%c	unsigned char	caractère
%s	char*	chaîne de caractères

FIGURE 3.3 – Exemple sur les formats d'impression en C

3.7.2 La fonction de saisie « scanf »

La fonction Scanf permet de saisir des données au clavier et de les stocker aux adresses spécifiées par les arguments de la fonctions.
scanf("chaîne de contrôle",argument1,...,argumentN)

3.8 Les instructions de contrôle

Le langage C dispose d'instructions de contrôle permettant de réaliser [19] :

3.8.1 Les instructions de branchements conditionnels

3.8.1.1 Choix :

Les instructions de branchements conditionnels de choix sont :

1. **L'instruction conditionnelle « if—else » :** L'instruction conditionnelle permet de tester une condition puis d'exécuter une action parmi deux actions possibles.
   Syntaxe :
   ```
   if ( expression)
     instructions1 ;
   else instructions2 ;
   ```
2. **Branchement multiple « switch » :** L'instruction switch est une instruction de choix multiple. Elle permet d'évaluer une expression puis d'exécuter une action parmi plusieurs actions étiquetées.
   Si la valeur de l'expression correspond à une des étiquettes, l'action correspondante est exécutée .
   Syntaxe :
   ```
   Switch ( expression)

     case constante1 : instructions ;
   ```

. . . . . .
case constante : instructions ;
. . . . . .
default : instructions ;

3.8.1.2 Boucles :

Les instructions de branchements conditionnels répétitives sont :

1. **Boucle « while » :** tant qu'une condition spécifiée n'est pas vérifiée, elle permet de répéter une ou plusieurs actions.
   Syntaxe :
   while (expression) instructions

2. **Boucle « do—while » :** Elle permet de répéter une ou plusieurs actions tant que la condition spécifiée n'est pas vérifiée.
   Syntaxe :
   Do instructions while (expression) ;

3. **Boucle « for » :** L'instruction for est une instruction de boucle faisant intervenir l'initialisation, le test de limite et l'incrémentation en une seule action.
   Syntaxe :
   for ([expr1] ; [expr2] ; [expr3])
   Instructions ;

3.8.2 Les instructions de branchements inconditionnels

Un branchement inconditionnel est utilisé pour passer une partie du script (en général, le branchement est utilisé dans une instruction de type branchement conditionnel). Par la suite, on ne reviendra pas à l'endroit où le programme a été quitté. [12]
Les instructions de branchement inconditionnel sont :

- ☐ goto
- ☐ break (associé aux boucles)
- ☐ continue (associé aux boucles)
- ☐ return

3.9 Boucle imbriqué

Une boucle imbriquée est une boucle dans une boucle, une boucle à l'intérieur du corps d'une autre boucle. Ce qui se passe est que le premier tour de la boucle externe déclenche la boucle interne, qui s'exécute jusqu'au bout. Puis le deuxième tour de la boucle externe déclenche la boucle interne une nouvelle fois. Ceci se répète jusqu'à ce que la boucle externe termine. Bien sûr, un break à l'intérieur de la boucle interne ou externe peut interrompre ce processus. [2]

3.10 segmentation entre code source

La segmentation entre code source est une extraction de structure et de contenu pour obtenir des entités textuelles plus petites à savoir [36] :

3.10.1 segmentation physique

La structure des codes sources composites en blocs homogènes (variable, méthode, commentaire,...). La segmentation physique du code source concerne la répartition spatiale de l'information du code source utilisant des méthodes, généralement très spécialisées pour l'analyse la structure physique du code source.

3.10.2 segmentation logique

La reconnaissance de l'organisation et l'étiquetage des blocs en diverses catégories constituent la phase d'identification de la structure. Elle est connue sous le nom de structuration logique du code source. La segmentation logique du code source, se rapporte au sens de cette organisation liée à l'interprétation de l'organisation des objets du code source.

3.11 Similarité des codes en programmation

Il existe différents types de la similarité [51] :

3.11.1 Similarité syntaxique

L'analyse syntaxique consiste à comparer les textes et la syntaxe (structure). Par exemple, une analyse simple consiste à comparer chacun des textes. Nous pouvons ensuite comparer les noms de méthodes, les appels, etc. en lisant le code du programme. De manière générale, nous pouvons connaître la similarité syntaxique de deux codes source en comparant chacune des caractéristiques du premier code source avec la même caractéristique d'un deuxième code source.

3.11.2 Similarité sémantique

Avec la similarité sémantique, on s'intéresse au sens (concept) caché derrière une méthode ou une variable. Ainsi, on regarde quels sont les concepts communs. Pour la similarité entre les codes sources, si deux unités représentent le même concept, alors ils sont plus similaires. Mais comme chaque développeur a ses propres habitudes de programmation et de nommage, il est difficile de déterminer la similarité par une simple comparaison du texte. C'est pourquoi l'analyse sémantique est intéressante.

3.12 Conclusion

Le code source en langage C, qui est généralement se caractérise par une forte structuration physique avec une organisation logique bien définie. Dans notre travail, on a besoins d'obtenir une représentation des documents textuels de type code source sous forme des blocs segmentés puis de faire une étude de la similarité sur ces segments.

Chapitre 4

Conception

4.1 Introduction

Dans ce chapitre, nous présenterons, La conception de notre système qui est une description logique de la façon dont le système va fonctionner. Elle consiste à façonner le Système et lui donner une forme et une architecture.

Nous avons donc essayé d'identifier les acteurs et les sous système avec les différentes étapes de La détection des documents électroniques code source plagiés au sein d'un corpus pour aider le correcteur à prendre les bonnes décisions. La conception du système proposé se présente sous forme d'une architecture globale ensuite, plus détaillé.

4.2 La conception du système

La section suivante insiste sur la présentation de notre système et donne une conception autour notre acteurs et sous système.

4.3 Architecture global du système

Ce schéma représente L'architecture générale de notre système se repose sur deux sous système :

- Le premier sous système « One by One ».
- Le deuxième sous système « One by Many ».

Chaque sous système est réalisé en un certain nombre d'étape :

1. Sélection du code source et cible.
2. Segmentation de structure du code source et cible.
3. Pré-filtrage et substitution du code source et cible.
4. Segmentation du code source et cible.
5. Analyse syntaxique et sémantique du code source et cible.

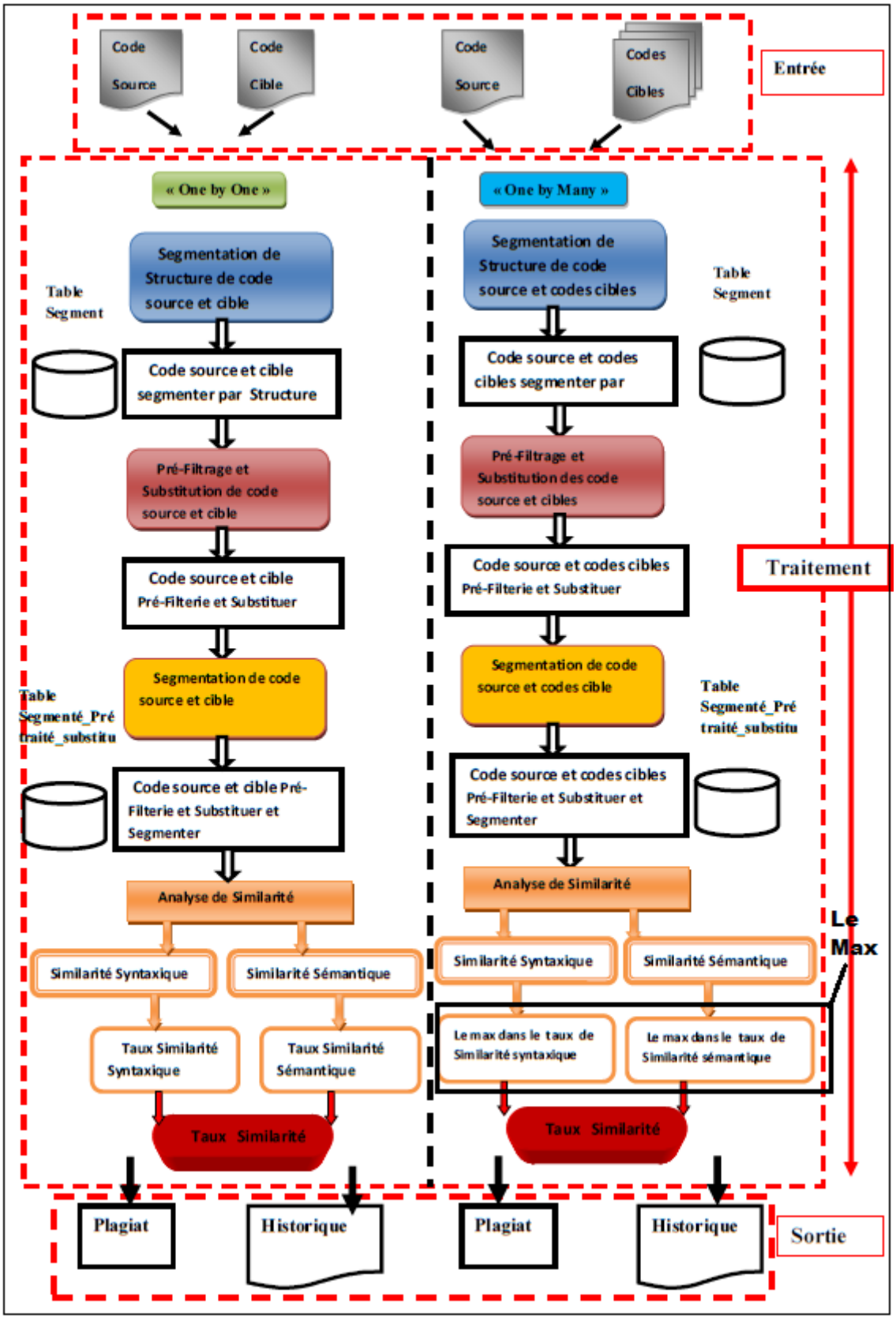


FIGURE 4.1 – Architecture générale de système de l'étude de la similarité des codes sources

4.4 Architecture détaillée du système

La section suivante présente l'architecture détaillée de notre système qui se compose par un espace d'administrateur, un enseignant, un sous système « One by One » et un sous système « One by Many ».

4.4.1 Administrateur

Dans notre système, l'administrateur est le responsable du système fait une fonction principale qui est la préparation de groupe. La préparation de groupe contient :

- L'insertion des étudiants.
- Le tri des étudiants par leurs moyennes.
- le choix de nombre de major dans un groupe.

On va faire une explication des fonctionnalité de l'acteur administrateur par :

1. **Unités fonctionnelles de l'administrateur :**

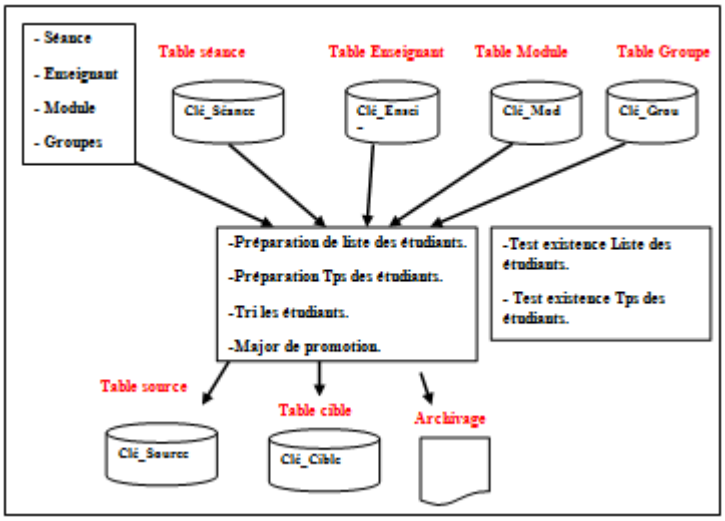


FIGURE 4.2 – Unités fonctionnelles de l'administrateur

2. **Diagramme de structure de l'administrateur :**

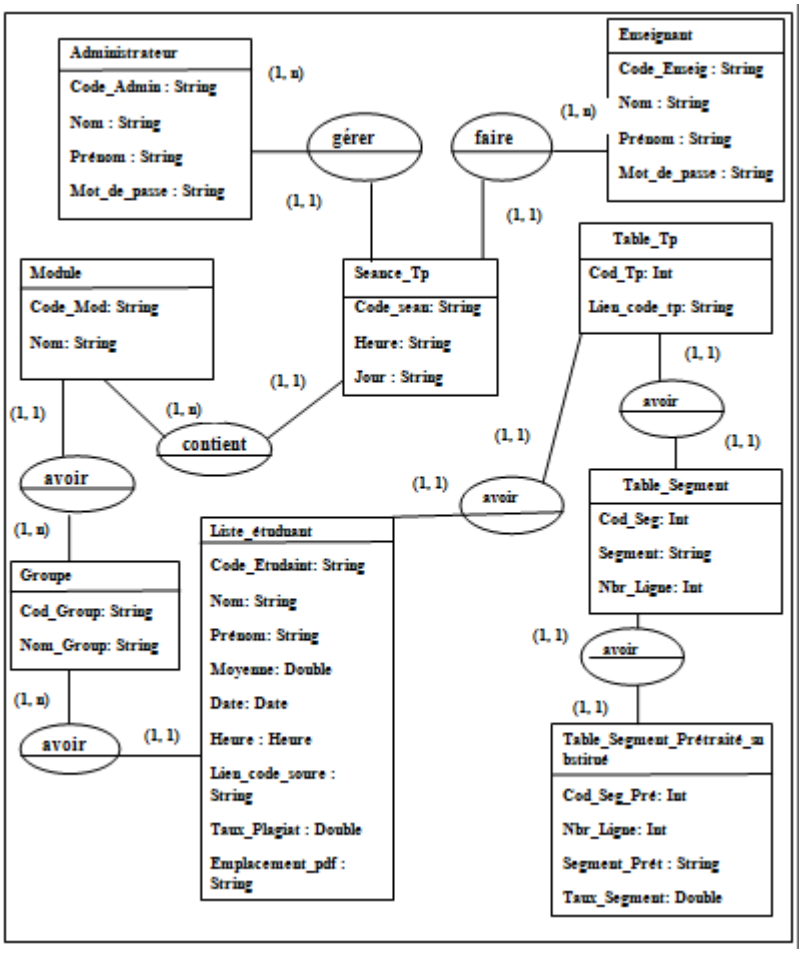


FIGURE 4.3 – Diagramme de structure de l'administrateur

4.4.2 Enseignant

Dans notre système, l'enseignant est un utilisateur qui fait deux fonctions principales qui sont :

1. **Exécution de l'analyse de la similarité :** L'exécution de l'analyse de la similarité fait par l'enseignant qui fait la sélection de code source, code cible, le choix de sous système « one by one» ou « one by many» puis fait l'analyse de la similarité.

2. **Évaluation :** L'évaluation des étudiants fait par la synthèse de résultat de taux plagiat.

On va faire une explication des fonctionnalité de l'acteur enseignant par deux diagrammes qui sont :

1. **unités fonctionnelles de l'enseignant :**

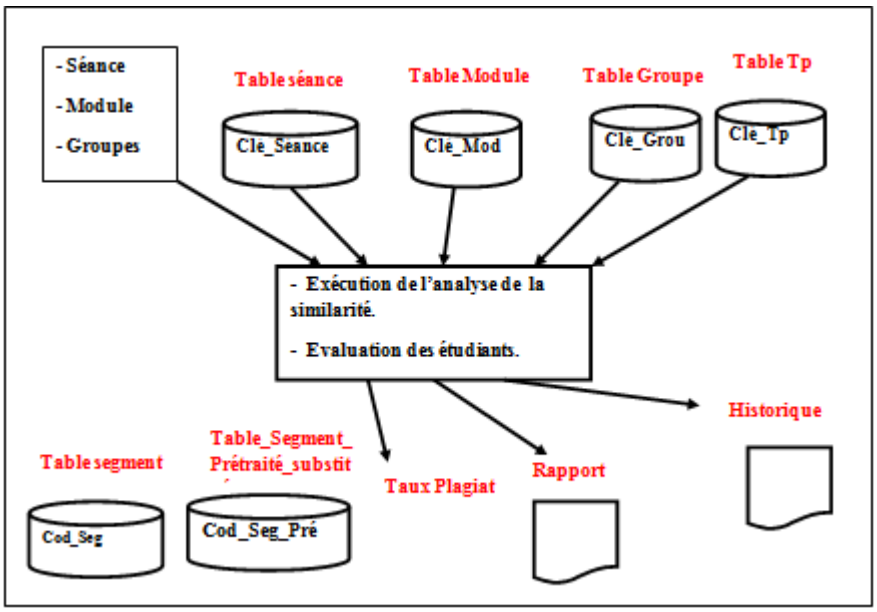


FIGURE 4.4 – Unités fonctionnelles de l'enseignant

2. **Diagramme de structure de l'enseignant :**

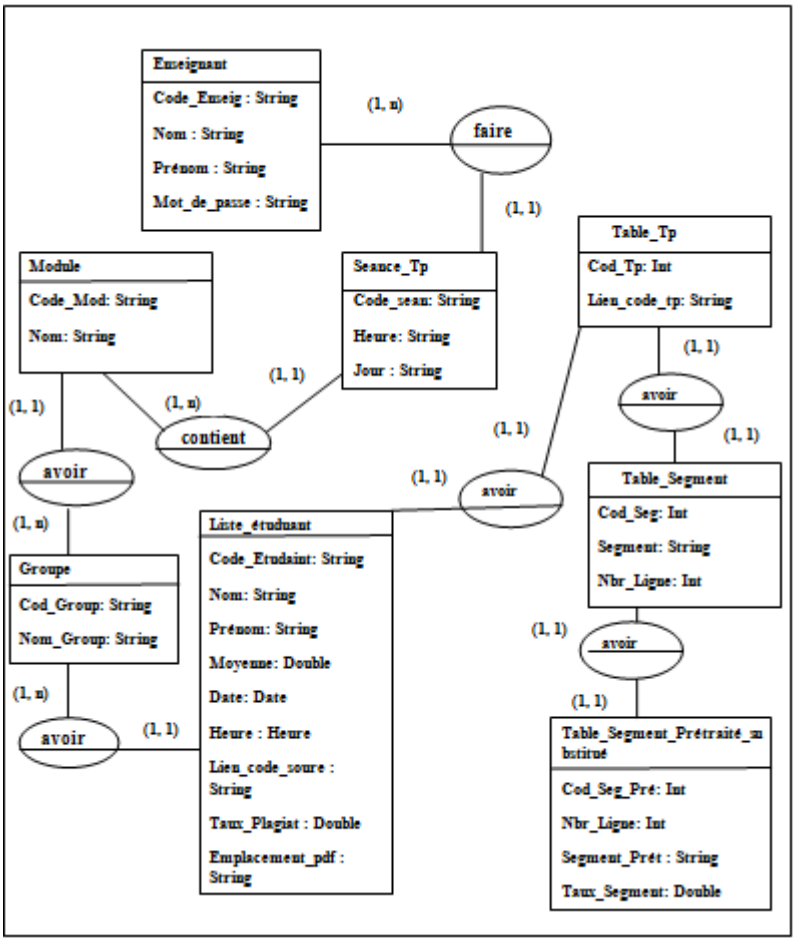


FIGURE 4.5 – Diagramme de structure de l'enseignant

4.4.3 sous système « One by One »

Le sous système « One by One » est fait la détection de plagiat d'un document code source contre un autre document code cible.

La section suivante présente la modélisation de sous système « One by One » sous forme d'une architecture globale puis une architecture détaillée.

4.4.3.1 Schéma globale de sous système « One by One »

Cette figure présente l'architecture globale de sous système « One by One ». Ce schéma montre toutes les étapes effectuées sur un document code source et un document code cible.

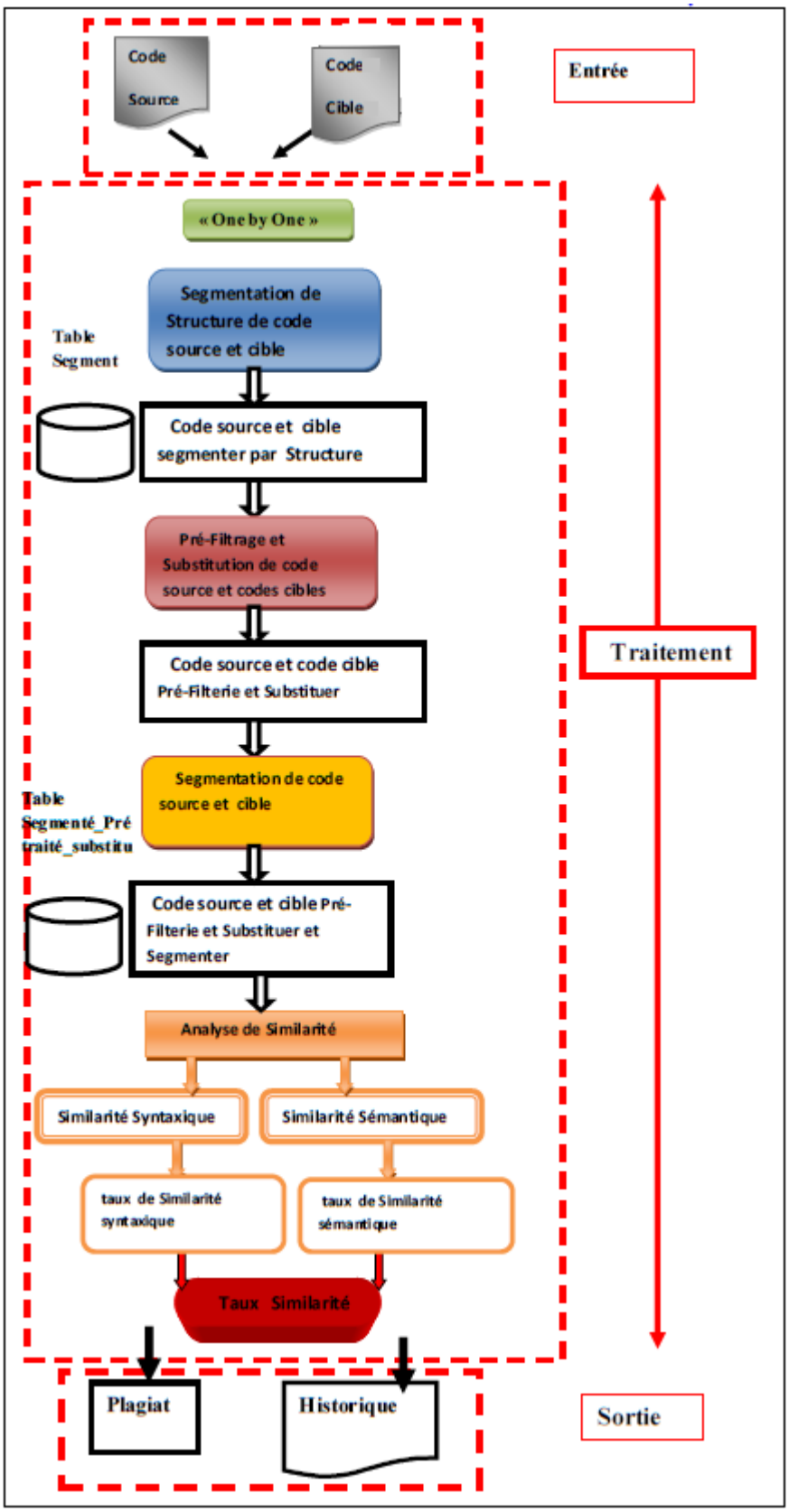


FIGURE 4.6 – Schéma générale de sous système « One by One »

4.4.3.2 Schéma détaillée de sous système « One by One »

La partie qui vient a pour but de donner les détails de chaque étape de la conception de notre sous système « One by One » avec l'entrée et la sortie de chaque étape puis le résultat

final.

- **Étape 01 : Sélection**
  L'étape de la sélection est l'étape d'entrée dans notre système.

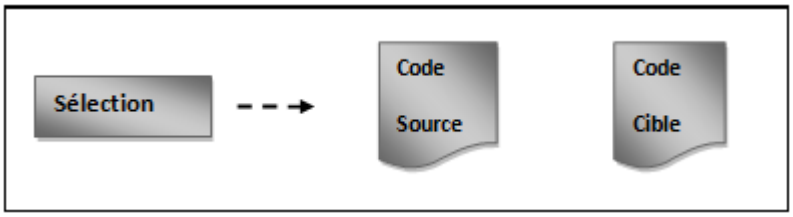


FIGURE 4.7 – Schéma de l'étape de la sélection « One by One »

Cette étape est constitué de choisir un document code source et un autre document code cible à partir d'une classe informatique.

- **Étape 02 : Segmentation de Structure**
  L'étape de la segmentation de structure est l'étape de sortie de l'étape de la sélection.

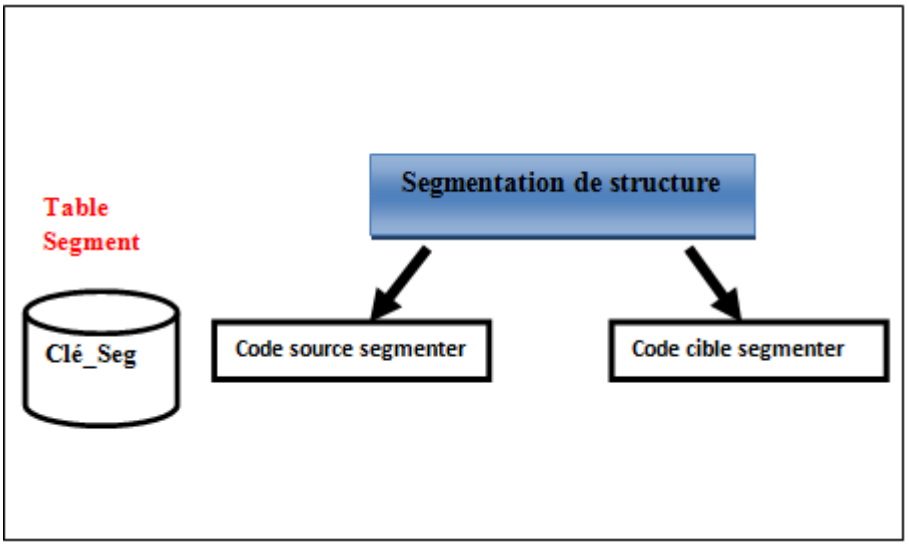


FIGURE 4.8 – Schéma de l'étape de segmentation de structure « One by One »

Cette figure présente l'étape de segmentation de la structure de code source et le code cible pour un rôle de faciliter la conservation de chaque partie de notre programme pour l'utiliser après.

Segmentation de la structure :

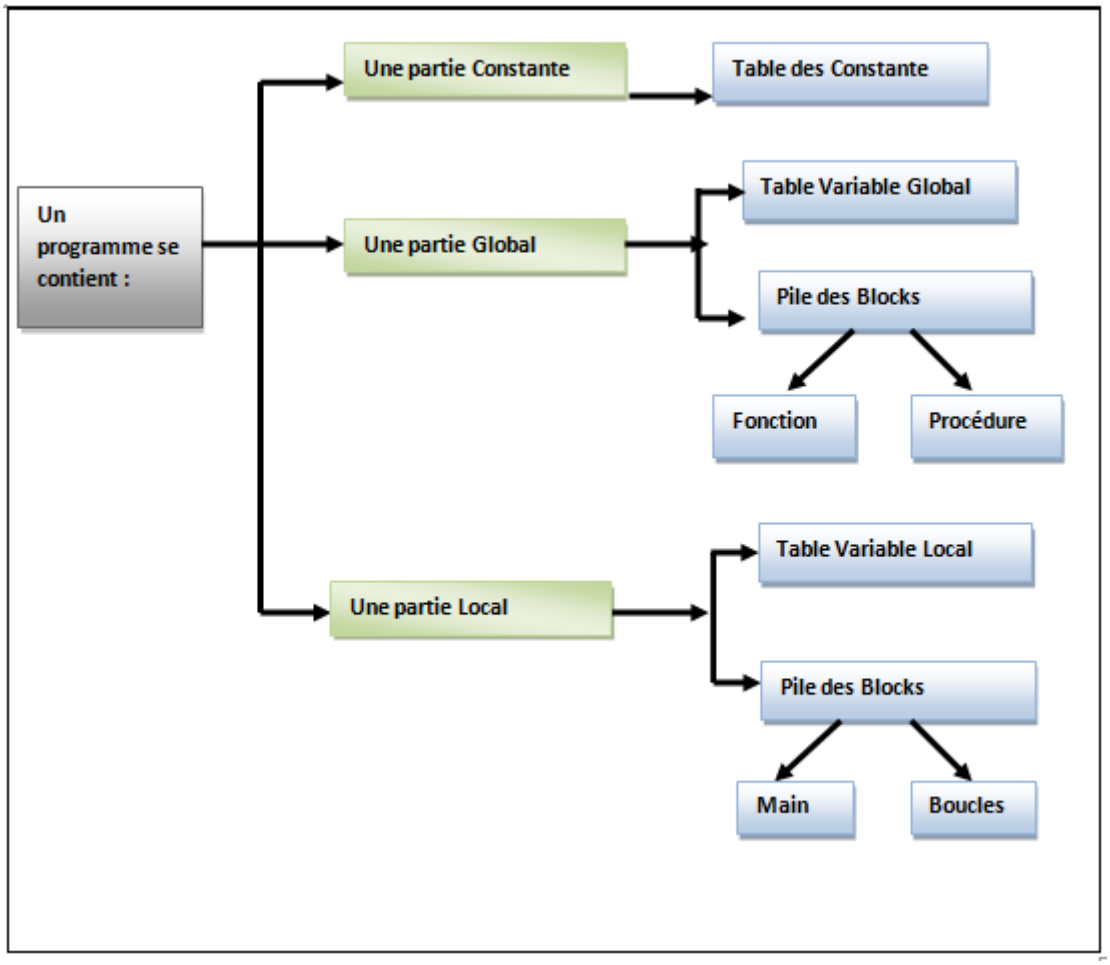


FIGURE 4.9 – Schéma d'étape de segmentation d'un programme informatique

Cette figure présente les composants d'un programme informatique comme une base dans la réalisation de la segmentation de la structure d' un code source et un code cible.

- **Étape 03 : Pré-filtrage et substitution**
  La segmentation de structure est l'étape d'entrée dans l'étape de pré-filtrage et substitution.

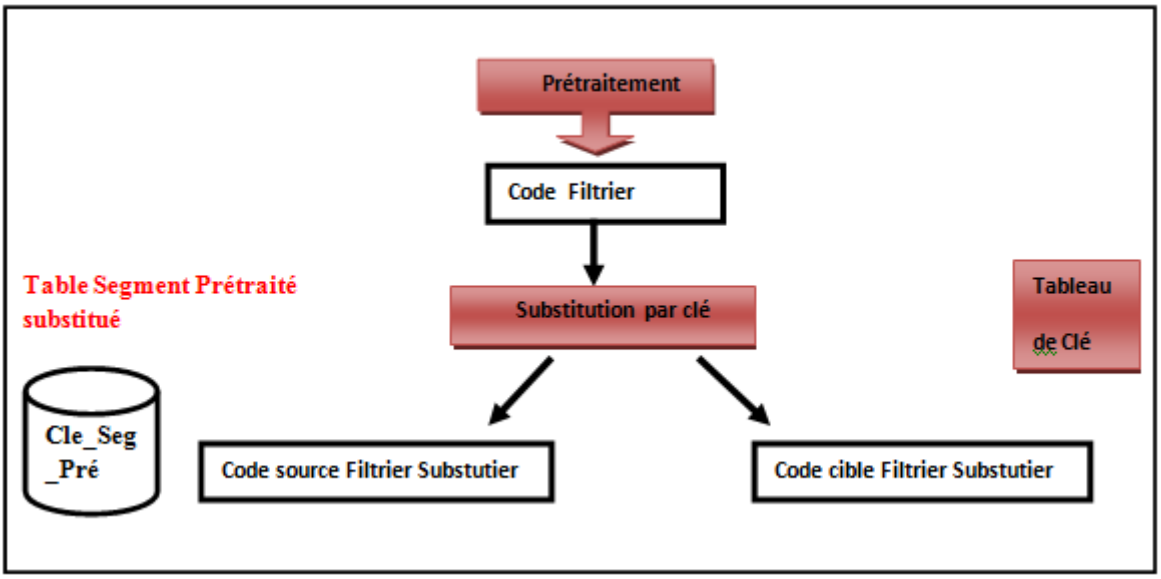


FIGURE 4.10 – Schéma de l'étape de Pré-traitement et substitution « One by One »

Cette figure explique l'étape de Pré-filtrage et substitution de code source et code cible qui ont besoin code segmenter pour obtenir à la fin un code filterie et substituer.

$a-$ **Pré-filtrage :** Cette opération a pour objectif de nettoyer le code source et le code cible, c'est-à-dire de réduire le jeu de caractères à manipuler afin de faciliter le travail. Le prétraitement consiste à enlever :

- Les lignes blanches.
- Les commentaires.
- Les caractères spéciaux par exemple les points virgules, les parenthèses, les espaces blancs, les adresses ....
- Concernant les entrés et les sorties (E/S), on s'intéresse par le variable à lire ou bien à afficher et l'opération (E/S) utiliser.
- Concernant boucle for on s'intéresse sauf par leur condition et on élimine l'initialisation et l'incrémentation.

$b-$ **Substitution :** Une substitution est une Opération d'identification et/ou de représentation le code source et code cible à l'aide d'un code. Notre codage numérique utilisé est exprimé dans le tableau ci-dessous :

Mot clé	Code
Procédure	Code_Procedure Code_nom 0000
Fonction	0000 Code_nom code_type
Variable	Code_type Code_nom
Opérateur	Leur indice dans la table des opérateurs
Block de boucle	Compteur des accolades
E/S	Compteur
Instruction	Remplacement

TABLE 4.1 – Tableau des codes du substitution

- **Étape 04 : Segmentation**
  L'étape de la pré-filtrage et substitution est l'étape d'entrée dans l'étape de segmentation.

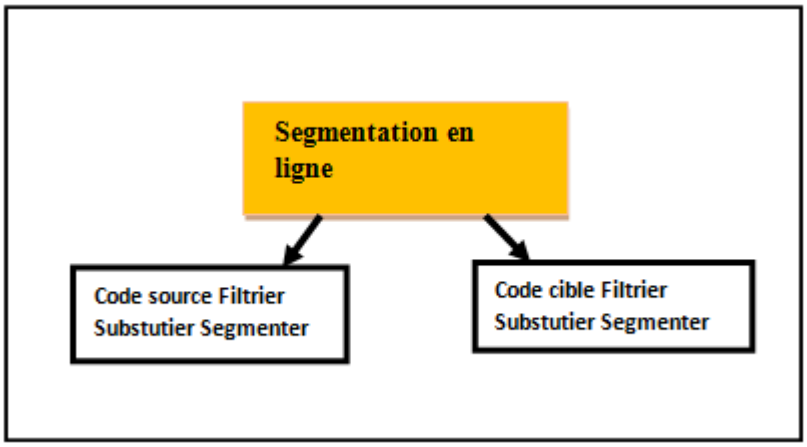


FIGURE 4.11 – Schéma d'étape de segmentation « One by One »

Cette Figure présente l'étape de segmentation de code source et code cible. On applique la segmentation en ligne sur les deux code filterie et substitué pour parvenir à les deux codes filterie, substituer et segmenter.

Segmentation en ligne : On appelle un corpus D et $|d|$ la taille du document code source d [33] .

- On note s[a ; b] = [s[a],. . . , s[b]] pour une chaîne de caractères s.
- Un segment est une section contigüe d'un document code source
- Un segment d'un document d est un élément (d ; p ; l) :
  p : la position du début du segment dans le document.
  l : sa longueur.
- Par conséquent, la concaténation texte (s1) . . . .texte (sm) des contenus des segments est égale au document initial.

Dans notre travail, on utilise la taille de segment est égal à $2^7 = 128$ bits.

- **Étape 05 : Analyse Similarité**
  Afin de faire la sélection, la segmentation de structure, le pré-filtrage, la substitution et la segmentation on arrive à l'étape de l'analyse de la similarité qui se compose par une partie d'analyse de la similarité syntaxique et une autre partie d'analyse de la similarité sémantique.

  $a-$ **Analyse de la similarité syntaxique :** L'étape de la segmentation est l'étape d'entrée dans l'étape de l'analyse de la similarité syntaxique.

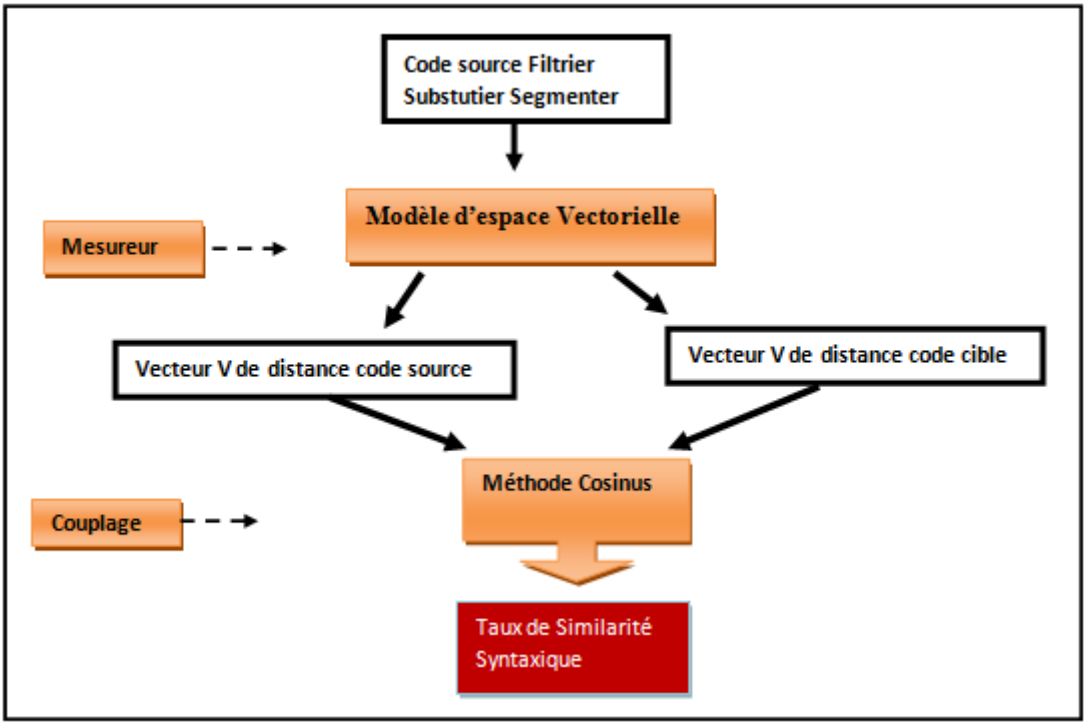


FIGURE 4.12 – Schéma de l'étape analyse de la similarité syntaxique « One by One »

Cette figure présente l'analyse de la similarité syntaxique qui accepte dans l'entrée un code source filtrier substutier segmenter et un code cible filtrier substutier segmenter pour retourner dans le sortie un taux d'analyse de la similarité syntaxique.
L'analyse de la similarité syntaxique se fait par :

- **Mesureur :** La Mesureur qui on utilise c'est le modèle vectorielle qui se déroule en deux étapes :
  (a) extraire les termes pertinents du document :
    - **TF**(Term Frequency).
    - **TF-IDF**(Term Frequency-Inversed Document Frequency).
  (b) calculer les poids des termes restants :
  $W_{ij} = TF_{ij} \times TFD_i$
- **Couplage :** La méthode de couplage qui on utilise dans notre approche c'est la méthode Cosinus, en tant que mesure de ressemblance entre deux documents code source et code cible.

$b-$ **Analyse de la similarité sémantique :**

Afin de faire l'analyse de la similarité syntaxique on arrive à l'étape de l'analyse de la similarité sémantique. Dans cette étape on applique la méthode de l'arbre sémantique et on fait une étude sur les techniques de transformation du code source en plus on applique une évaluation de la robustesse des séquences structurelles. [48]
Notre arbre sémantique est comme suit :

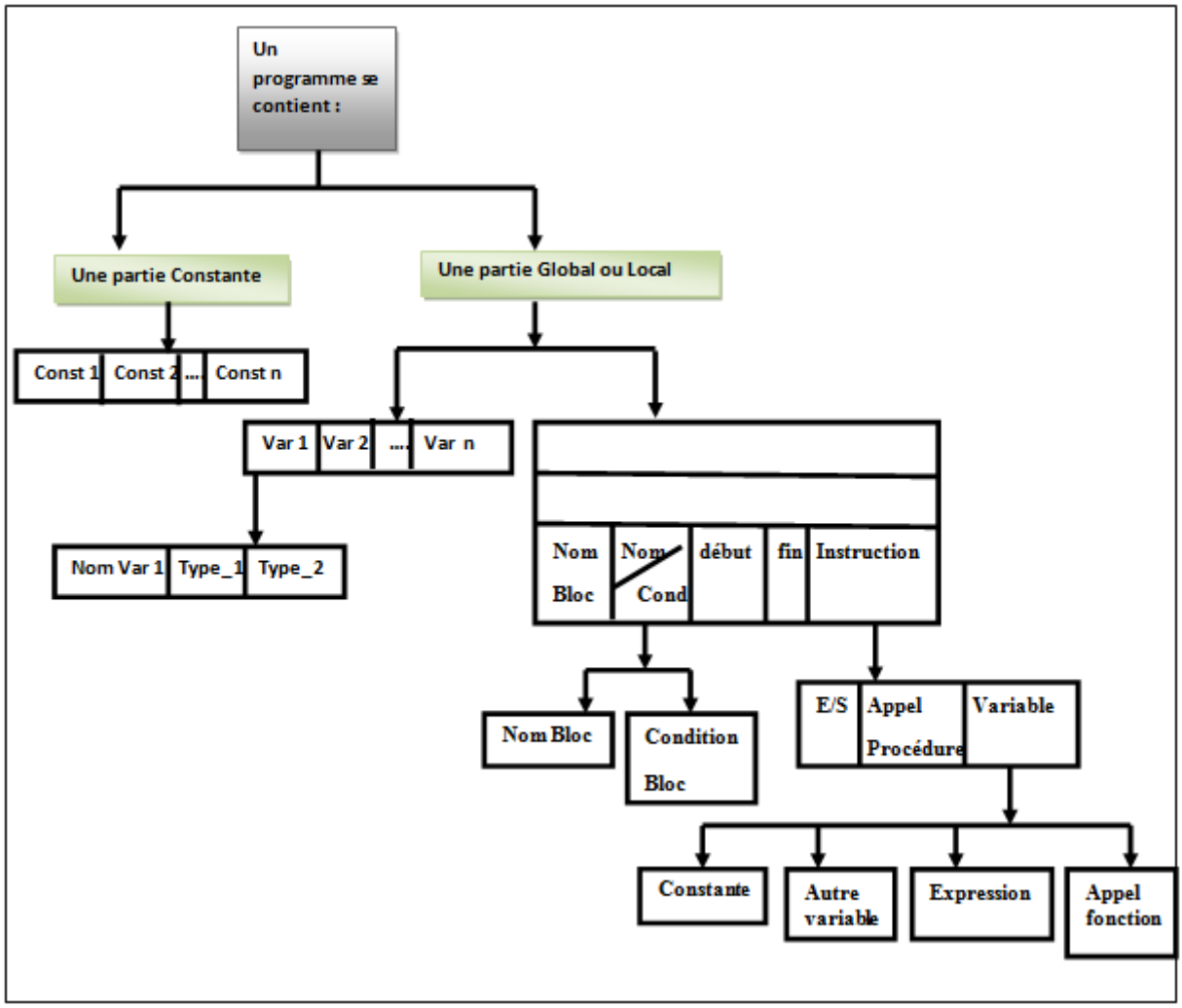


FIGURE 4.13 – Processus de construction de l'arbre sémantique dans un programme informatique

Mesureur de techniques de transformation du code source :

Il existe différentes techniques de transformation du code source qui sont souvent utilisées lors de opérations de plagiat.

Ces techniques permettant de différencier le contenu d'un code plagié par rapport à celui du code original tout en conservant les mêmes fonctionnalités d'origines Nous pouvons distinguer deux types de transformations :

$a-$ Les transformations du premier type sont de nature lexicale, parmi ces transformations nous citons :

- Attribution des nouveaux noms aux identifiants (variables, fonctions).
- La substitution des chaînes de caractères constantes par des chaînes de codes (code Ascii, Unicode, etc) tel que le contenu soit conservé.
- La modification des Commentaires : dans notre cas on élimine les commentaires.

$b-$ Les transformations du second type sont de nature structurelle nécessitant une connaissance du langage et une forte dépendance à la grammaire qui le définit. Parmi les transformations structurelles les plus couramment utilisées nous citons :

- Le changement de l'ordre des blocs d'instructions, de tel sorte que le comportement du programme ne soit pas affecté.
- La réécriture des expressions (permutation entre les opérandes et les opérateurs).
- Le changement du type des variables.
- L'ajout redondant d'instructions, de blocs d'instructions ou de variables, à condition que le comportement du programme ne soit pas modifié.
- La dégénérescence du flux de contrôle.
- La substitution des structures de contrôle itératives ou conditionnelles par d'autres structures de contrôle équivalentes.
- La substitution des appels de fonctions par les corps de ces fonctions.

Coupleur d'évaluation de la robustesse des Séquences Structurelles :

Dans cette partie nous évaluons la robustesse des Séquences Structurelles vis-à-vis des différentes techniques de plagiats qui tentent de rendre le code illisible et de le différencier de l'original. Ces techniques ont étaient classées en six niveaux par Faidhi et Robinsons. [48]

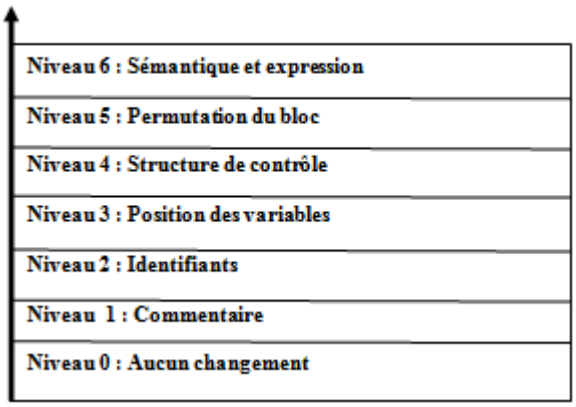


FIGURE 4.14 – Les niveaux du spectre des techniques de plagiat du code source

Les modifications effectuées sur le code original sont comme suit :

- **Niveau 0 :** Aucune modification.
- **Niveau 1 :** Modification des commentaires, ajout de nouveaux commentaires, suppression de commentaires et modification des chaînes de caractères dans les messages de sortie.
- **Niveau 2 :** Changements des noms de variables + les changements du niveau 1.
- **Niveau 3 :** Changements des déclarations et de leur position dans le code (remplacer deux constantes par deux nouvelles variables déclarées, changement des positions de déclaration entre trois variables) + les changements du niveau 2.
- **Niveau 4 :** Remplacer deux blocs itératifs "For" par deux blocs "While", et un block itératif "While" par un block "For" + les changements du niveau 3.
- **Niveau 5 :** Changement de la modularité (création de deux nouvelles fonctions, changement de position entre deux fonctions existantes) + les changements du niveau 4.

– **Niveau 6 :** Changements de deux expressions logiques et permutation entre le contenu du block "If" et "Else" en modifiant l'expression d'évaluation du test "If" + les changements du niveau 5.

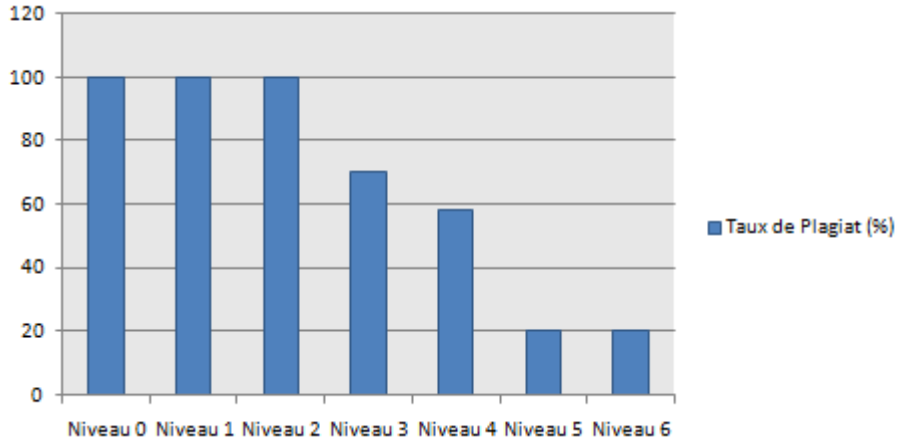


FIGURE 4.15 – Robustesse des séquences structurelles aux différents niveaux de plagiat

4.4.4 sous système « One by Many »

Le sous système « One by Many » est fait la détection de plagiat d'un document code source contre des groupes des documents code cible.

Le sous système « One by Many » est une optimisation de sous système « One by One». La section suivante présente la modélisation de sous système « One by Many »sous forme d'une architecture globale puis une architecture détaillée.

4.4.4.1 Schéma globale de sous système « One by Many »

Cette figure présente l'architecture globale de sous système « One by Many ». Cette schéma montre toutes les étapes effectuées sur un document code source et des groupes des documents codes cible.

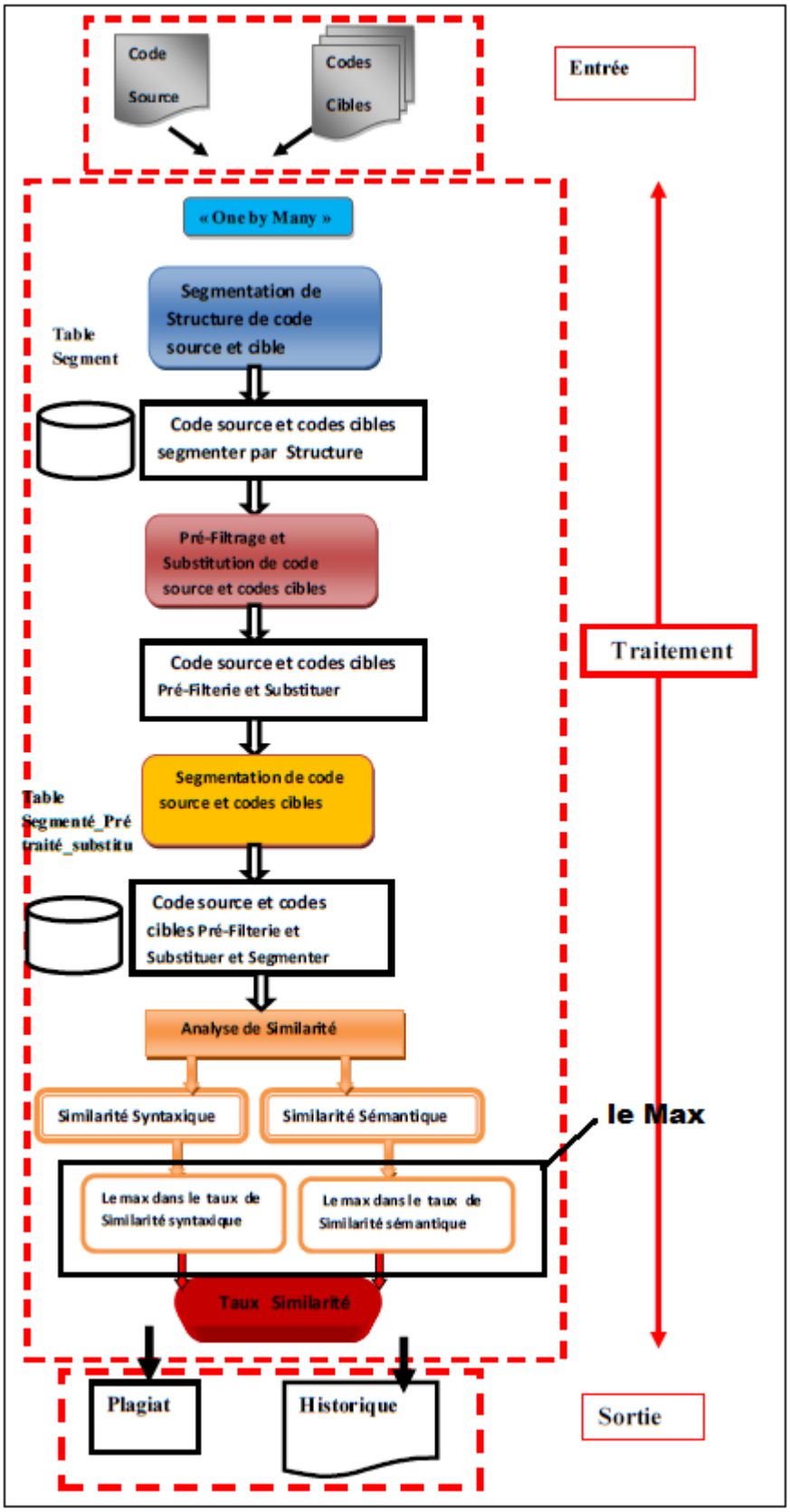


FIGURE 4.16 – Schéma générale de sous système « One by Many »

4.4.4.2 Schéma détaillé de sous système « One by Many »

La partie qui vient a pour but de donner les détails de chaque étape de la conception de notre sous système « One by Many » avec l'entrée et la sortie de chaque étape puis le résultat

final.

- **Étape 01 : Sélection**

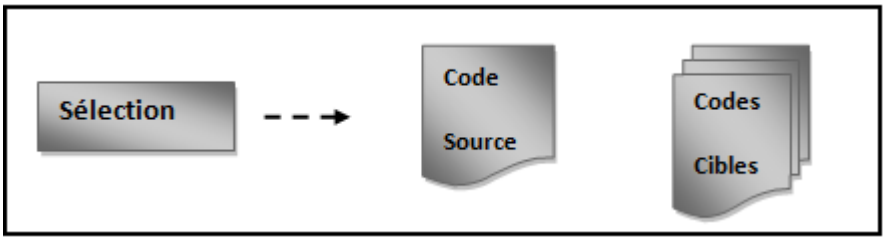


FIGURE 4.17 – Schéma de l'étape de sélection « One by Many »

Dans cette étape, on fait la sélection d'un code source d'un enseignant et un certain nombre de groupe des étudiants (File de code cible).

- **Étape 02 : Segmentation de Structure**

Dans cette étape, On fait la segmentation de structure de code source et tous les codes cibles. La segmentation de structure est déjà présentée.

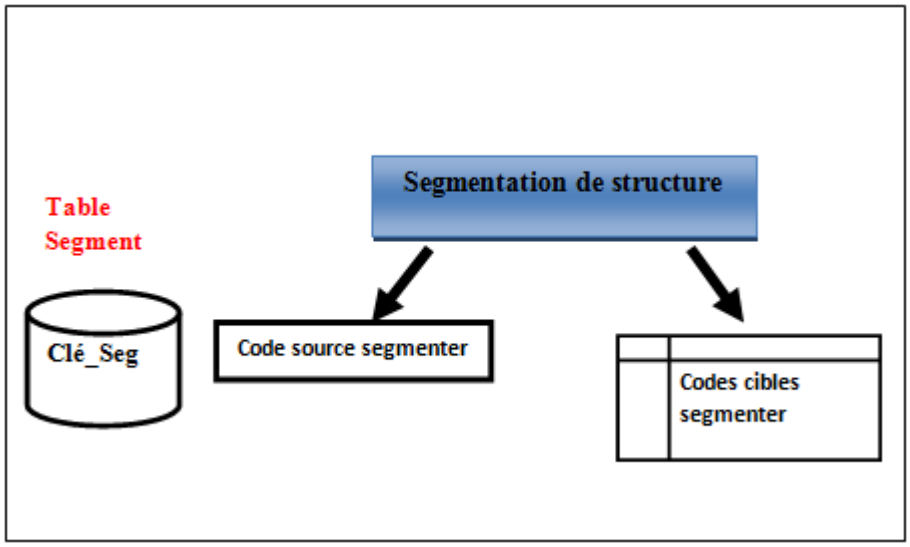


FIGURE 4.18 – Schéma de l'étape de segmentation de structure « One by Many »

- **Étape 03 : Pré-filtrage et substitution**

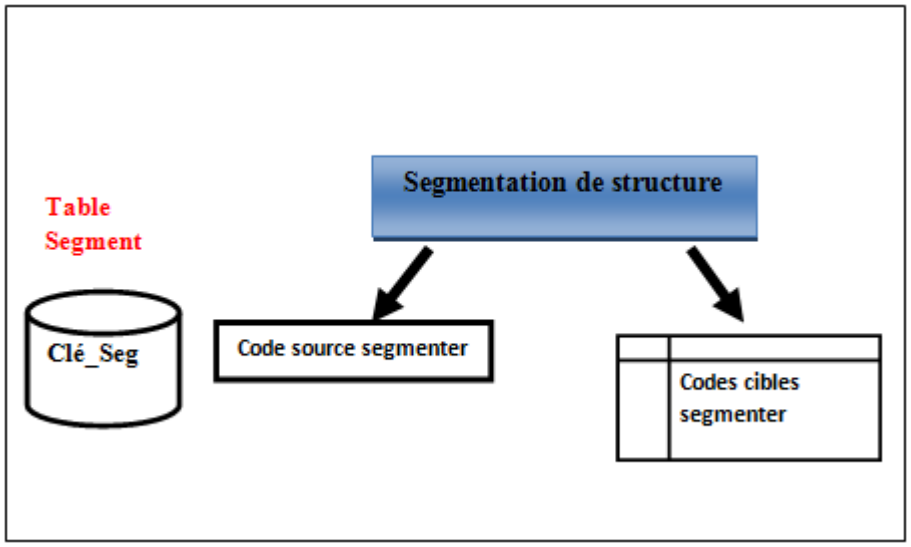


FIGURE 4.19 – Schéma de l'étape de Pré-traitement et substitution « One by Many »

Cette figure représente l'étape de Pré-filtrage et substitution d'un code source et des groupes des codes cible .

$a-$ **Pré-Filtrage :** déjà définie dans la partie 4.4.3.2.

$b-$ **Substitution :** déjà définie dans la partie 4.4.3.2.

- **Étape 04 : Segmentation**

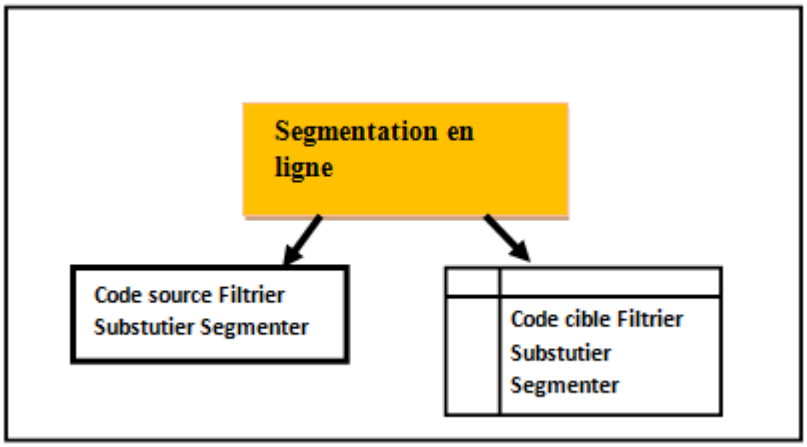


FIGURE 4.20 – Schéma de l'étape de segmentation « One by Many »

Cette Figure présente l'étape de segmentation de code source et des groupes de codes cible. On applique la segmentation en ligne pour parvenir à un code source filterie substitué segmenté et des groupes des codes cibles filteries substitués segmentés.

- **Étape 05 : Analyse similarité**

Afin de faire la sélection, la segmentation de structure, le pré-filtrage, la substitution et la segmentation on arrive à l'étape de l'analyse de la similarité qui se compose par une partie d'analyse de la similarité syntaxique et une autre partie d'analyse de la similarité sémantique.

$a-$ **Analyse de la similarité syntaxique :**

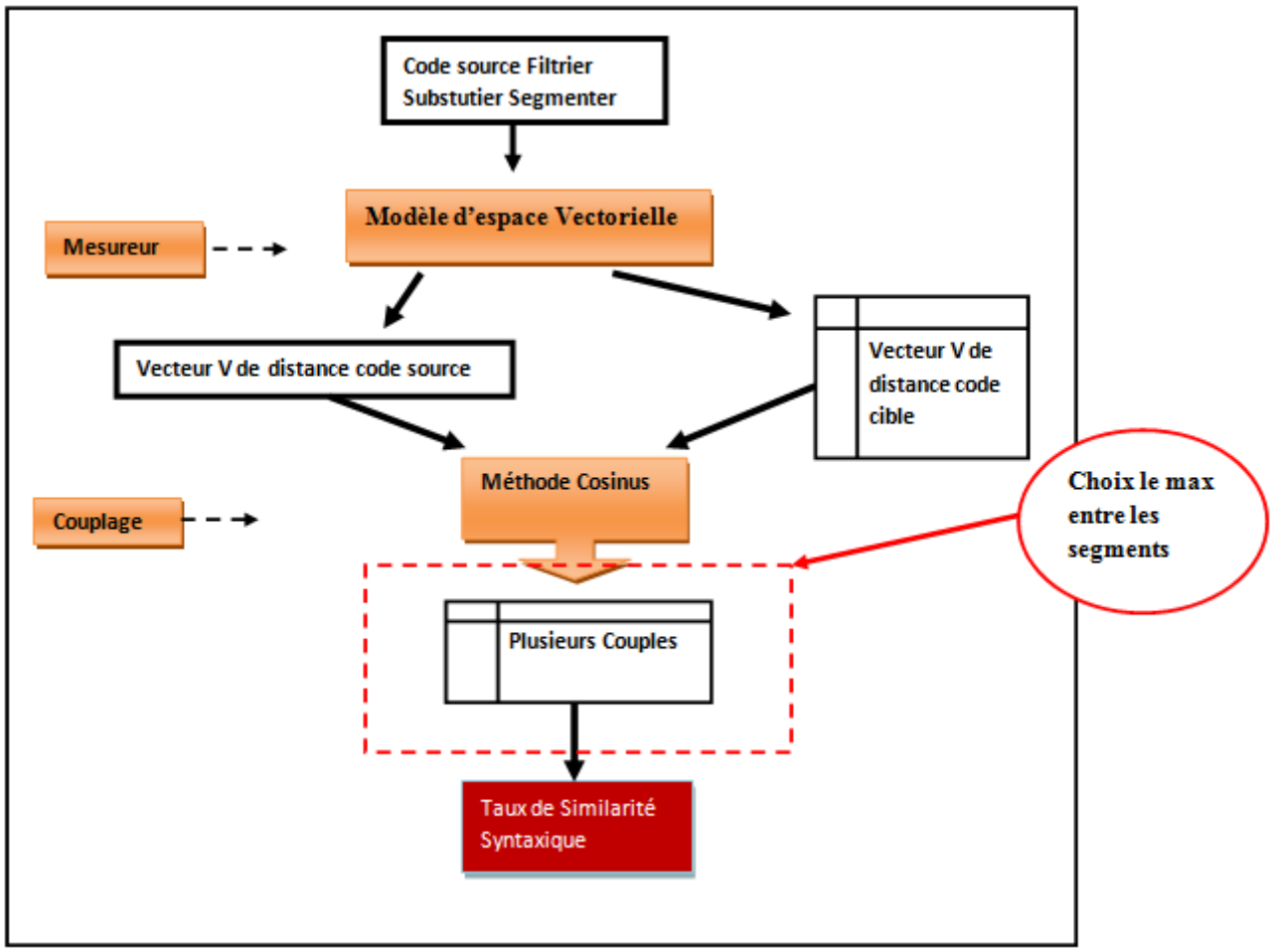


FIGURE 4.21 – Schéma de l'étape analyse de la similarité syntaxique « One by Many »

Cette figure présente l'analyse de la similarité syntaxique qui accepte dans l'entrée un code source filtrier substutier segmenter et des groupes des codes cible filtrier substutier segmenter pour retourner dans le sortie un taux d'analyse de la similarité syntaxique.

- **Mesureur :** définie dans la partie 4.4.3.2.
- **Couplage :**déjà définie dans la partie 4.4.3.2 .

$b-$ **Analyse de la similarité sémantique :** déjà définie dans la partie 4.4.3.2 .

4.5 Conclusion

Ce chapitre a donné une vision sur notre travail, et a donné l'aspect conceptuel de l'application qui fait l'étude de la similarité dans les documents électroniques administratifs les codes source en langage C.

Nous avons commencé par une architecture globale de notre système puis on a essayé d'expliquer une conception détaillée des différents sous système. Notre système est composé par deux acteurs qui sont un administrateur et un enseignant en plus par un sous système « One by One » et un autre sous système « One by Many », ici, on a clarifié ce point sous forme globale et détaillée et on a indiqué leurs étapes de réalisation.

Chapitre 5

Implémentation

5.1 Introduction

Nous arrivons dans ce chapitre à la description de l'aspect pratique de notre travail. Nous commençons par expliquer l'environnement de développement ainsi que les outils permettant l'implémentation de notre travail : le langage Java, l'environnement Eclipse pour la réalisation de l'application et Xampp Server pour la manipulation de la base de donnée relationnelle.

Nous passons, ensuite, à la description des différentes étapes du processus d'étude de la similarité dans les codes source illustrant chaque étape par un ensemble des interfaces.

5.2 Présentation du langage d'application

L'application a été développée en JAVA pour lequel nous avons opté car il est de plus en plus utilisé dans le monde de la recherche scientifique. En effet, ce langage de programmation présente un large avantage car les programmes peuvent être exécutés sur différents systèmes d'exploitation et architectures matérielles.

La section suivant présente les outils logiciels utilisés pour la réalisation de cette application.

1. **Présentation Eclipse :** est un IDE, *Integrated Developmen Environment* (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation proposant un certain nombre de raccourcis et d'aide à la programmation. L'environnement Eclipse a beaucoup d'avantages, on cite comme exemple [31] :

   - Plate-forme ouverte pour le développement d'applications et extensible grâce aux plug-ins ;
   - Support multi-langages, multi-OS (Win, Linux, Mac) ;
   - Très rapide à l'exécution ;
   - Nombreuses fonctionnalités proposées par le JDT (Java Development Tool) ;
   - Historique local des dernières modifications réalisées.

   La figure suivante représenter le logo d'eclipse IDE.

FIGURE 5.1 – Logo d'eclipse IDE

2. **Xampp Serveur :** est un ensemble de logiciels permettant de mettre en place facilement un serveur Web et un serveur FTP. Il s'agit d'une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.

   Parmi les logiciels libres qui contiennent XAMPP on a MySQL qu'est un langage d'interrogation de bases de données, supporté par la plupart des systèmes de gestion de bases de données relationnelles. [24]

5.3 Présentation de l'application

5.3.1 Fenêtre d'accueil

La figure suivante représente la fenêtre d'accueil de notre système.

FIGURE 5.2 – Fenêtre d'accueil

Quand l'utilisateur clique sur le bouton démarrer une fenêtre va être apparaitre pour choisir est-ce que c'est un :

1. administrateur
2. enseignant

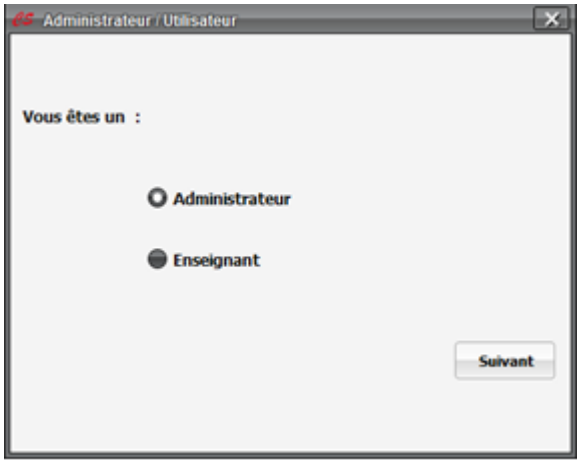


FIGURE 5.3 – Fenêtre de choisir l'utilisateur de système

5.3.2 Administrateur

L'administrateur est le responsable de système qui fait des fonctions principales, la section suivante montre les opérations possibles par l'administrateur.

Cette fiche présente la fenêtre d'authentification de l'administrateur.

FIGURE 5.4 – Fenêtre d'authentification

après l'administrateur fait l'authentification il apparait leur fenêtre principale comme la figure ce-dessous.

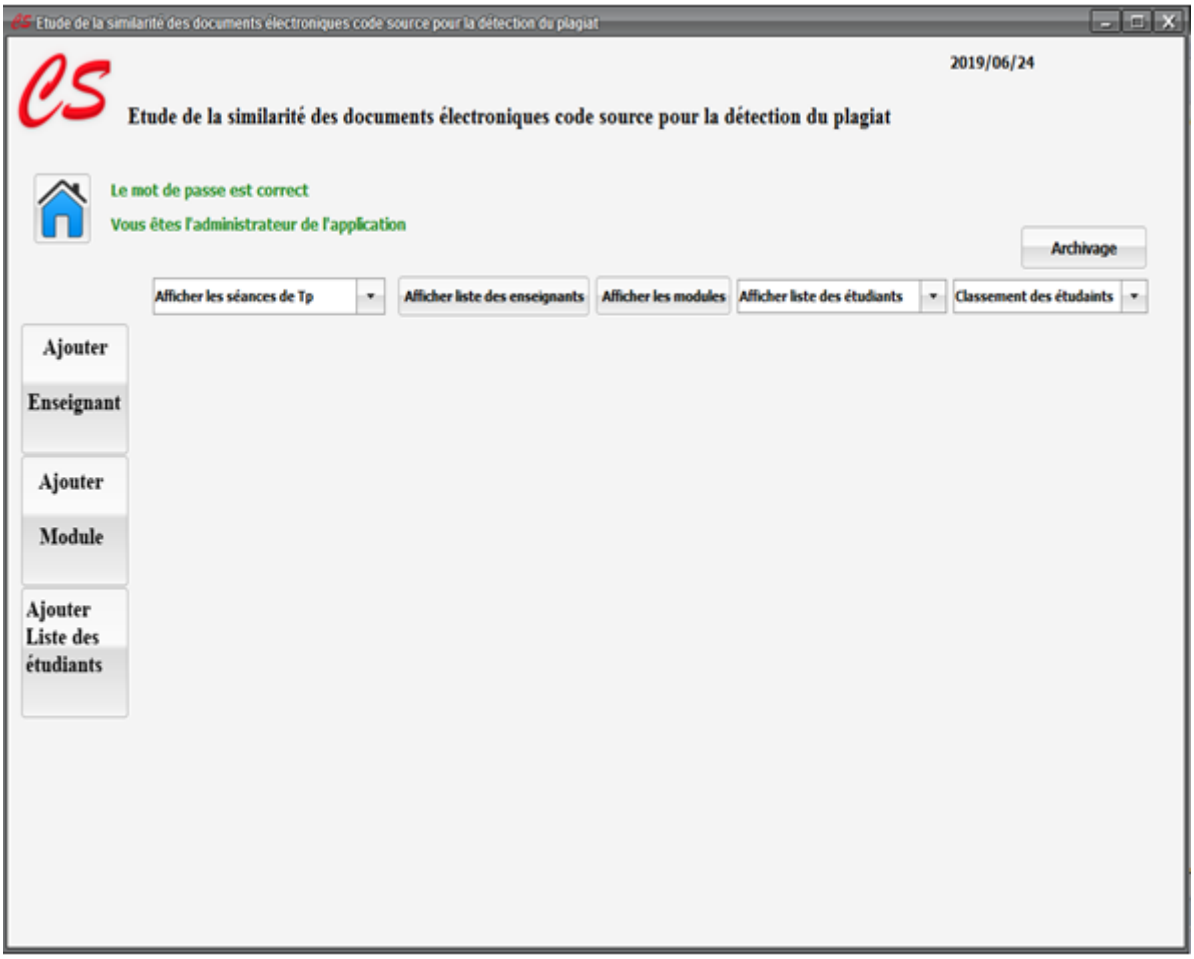


FIGURE 5.5 – Fenêtre de l'administrateur

la fenêtre de l'administrateur donne les possibilités de :

1. Ajouter enseignant.
2. Ajouter module.
3. Ajouter liste des étudiants.
4. Faire un classement des étudiants selon leur moyenne.
5. Afficher Les séances Tps existant.
6. Afficher les enseignants.
7. Afficher liste des étudiants.

En plus, La fenêtre de l'administrateur contient un bouton archivage qui permet d'afficher toutes les opérations faites dans le système soit par un administrateur ou bien par un enseignant.
Exemple sur les opérations de l'administrateur :
L'ajout d'un enseignant : Cette fiche montre l'ajout d'un enseignant.

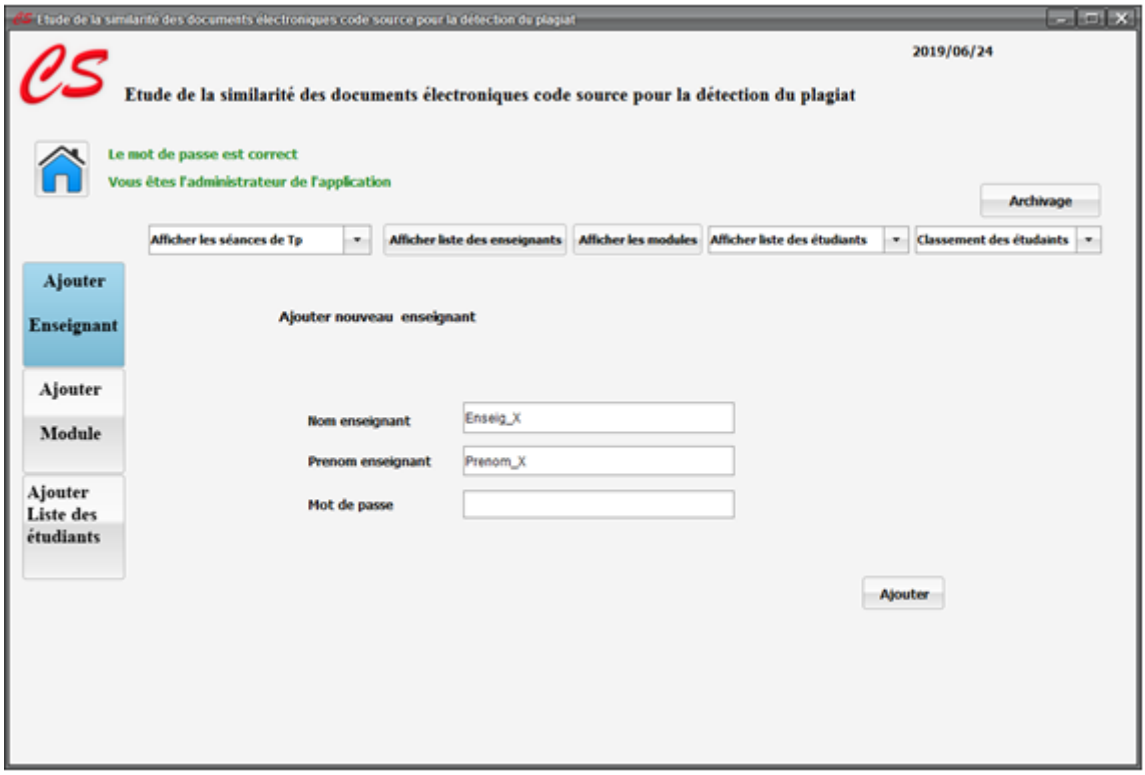


FIGURE 5.6 – Fenêtre d'ajout d'un enseignant

Les informations de l'enseignant insérer par l'administrateur va stocker dans la base de données dans la table d'enseignant.

Code_e	Nom	Prenom	mot_de_passe	Cod_s
1	Enseig_X	Prenom_X	mot_1	1
2	Enseig_Y	Prenom_Y	mot_2	1
3	Enseig_Z	Enseig_Z	mot_3	2

FIGURE 5.7 – Table enseignant

L'ajout d'un module : Cette fiche montre l'ajout d'un module.

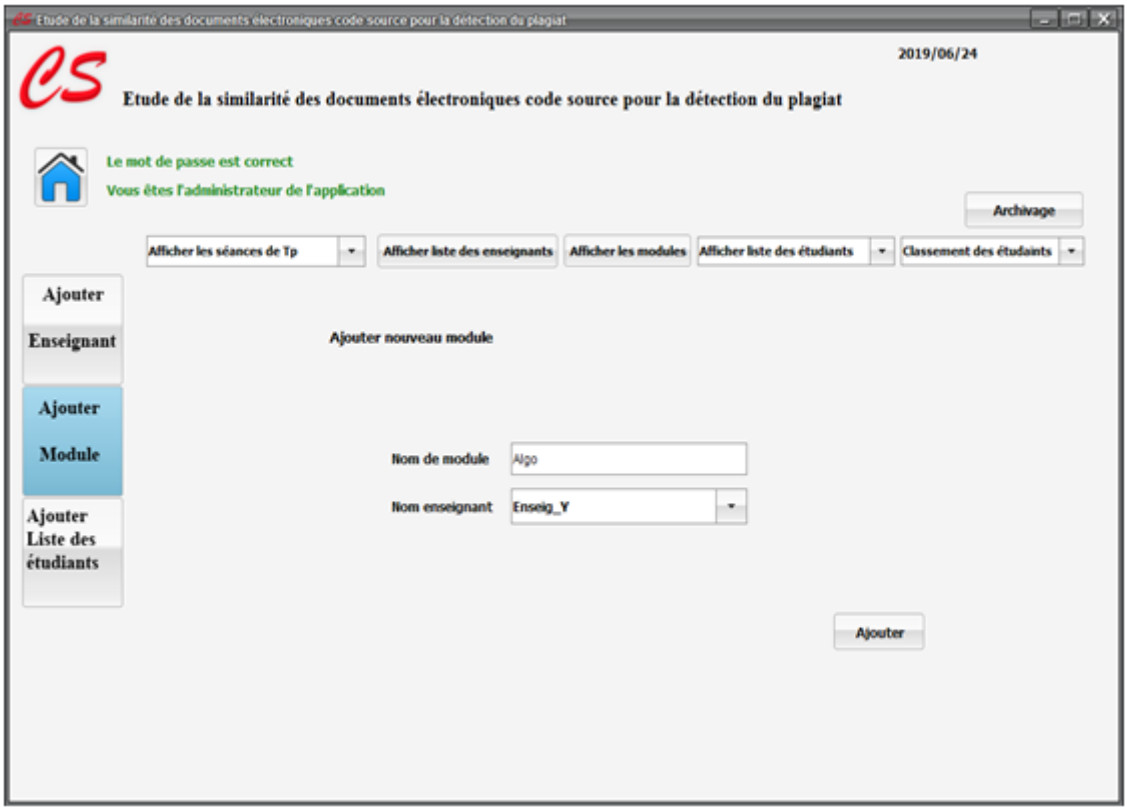


FIGURE 5.8 – Fenêtre d'ajout d'un module

Les informations de module insérer par l'administrateur va stocker dans la base de données dans la table module.

code_m	Nom_module	Nom_enseig
1	OMI	Enseig_X
2	Algo	Enseig_Y
3	PL	Enseig_Z

FIGURE 5.9 – Table module

L'ajout d'une liste des étudiants : Cette fenêtre présente l'ajout d'une liste des étudiants dans un groupe, cette opération demande :

- Le choix de nom de groupe.
- La liste des étudiants d'après une délibération finale contient leurs moyennes annuelles.
- Les travaux pratiques des étudiants en langage C (.c).

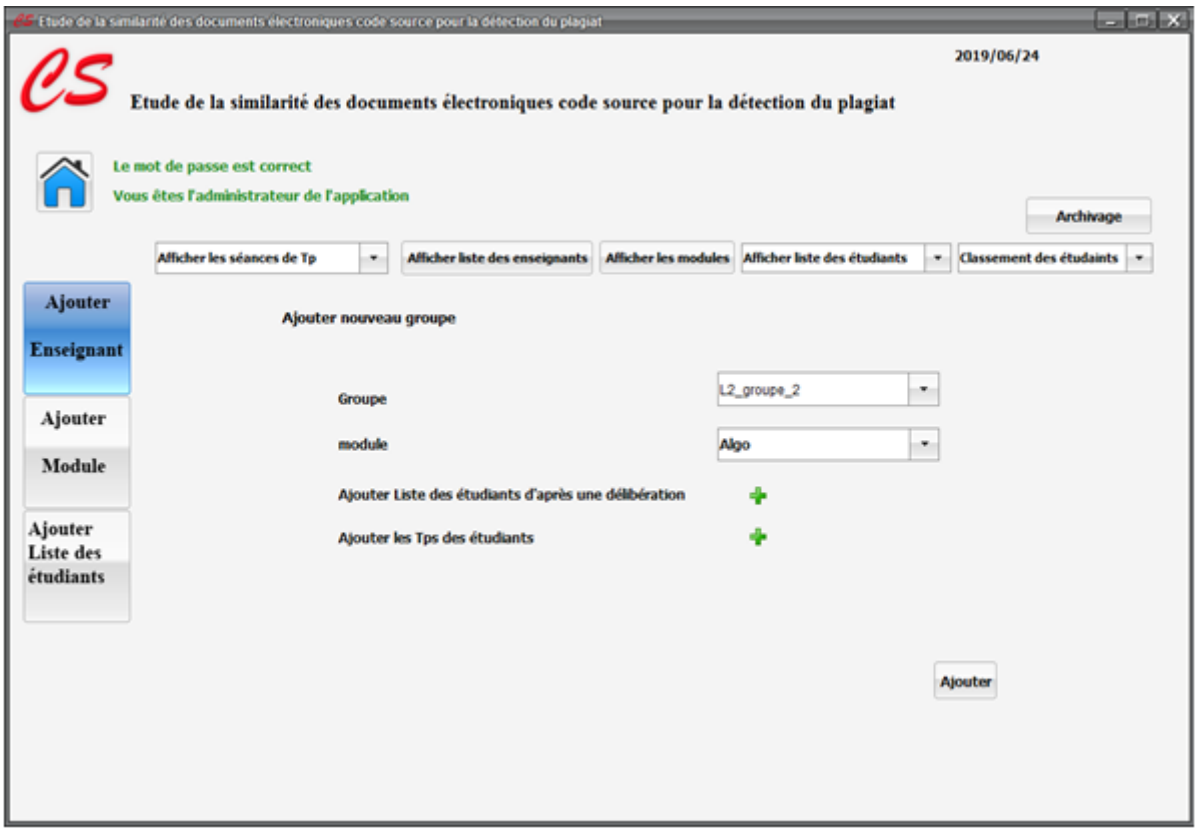


FIGURE 5.10 – Fenêtre d'ajout d'une liste des étudiants

La liste des étudiants va stocker dans la base de données.

L2_grp_2	code_L2_Groupe2	Nom	Prenom	Moyenne	Date	Heure	Emplacement_code	Nombre_ligne_seg	Nombre_ligne_seg_prét	Plagiat	Emplacement_pdf
2	14/3501	X1	Y1	8	NULL	NULL	G:\Tp_Algo\main_X1.c	NULL	NULL	NULL	NULL
2	14/35010	X10	Y10	12	NULL	NULL	G:\Tp_Algo\main_X10.c	NULL	NULL	NULL	NULL
2	14/35011	X11	Y11	15.5	NULL	NULL	G:\Tp_Algo\main_X11.c	NULL	NULL	NULL	NULL
2	14/35012	X12	Y12	8.5	NULL	NULL	G:\Tp_Algo\main_X12.c	NULL	NULL	NULL	NULL
2	14/35013	X13	Y13	10	NULL	NULL	G:\Tp_Algo\main_X13.c	NULL	NULL	NULL	NULL
2	14/35014	X14	Y14	10	NULL	NULL	G:\Tp_Algo\main_X14.c	NULL	NULL	NULL	NULL
2	14/35015	X15	Y15	6	NULL	NULL	G:\Tp_Algo\main_X15.c	NULL	NULL	NULL	NULL
2	14/3502	X2	Y2	14	NULL	NULL	G:\Tp_Algo\main_X2.c	NULL	NULL	NULL	NULL
2	14/3503	X3	Y3	12.5	NULL	NULL	G:\Tp_Algo\main_X3.c	NULL	NULL	NULL	NULL
2	14/3504	X4	Y4	13	NULL	NULL	G:\Tp_Algo\main_X4.c	NULL	NULL	NULL	NULL
2	14/3505	X5	Y5	5.5	NULL	NULL	G:\Tp_Algo\main_X5.c	NULL	NULL	NULL	NULL
2	14/3506	X6	Y6	11	NULL	NULL	G:\Tp_Algo\main_X6.c	NULL	NULL	NULL	NULL
2	14/3507	X7	Y7	8	NULL	NULL	G:\Tp_Algo\main_X7.c	NULL	NULL	NULL	NULL
2	14/3508	X8	Y8	11	NULL	NULL	G:\Tp_Algo\main_X8.c	NULL	NULL	NULL	NULL
2	14/3509	X9	Y9	14	NULL	NULL	G:\Tp_Algo\main_X9.c	NULL	NULL	NULL	NULL

FIGURE 5.11 – Table de liste des étudiants

Tri liste des étudiants : Le tri de liste des étudiants c'est-à-dire faire un ordre sur étudiants selon leurs moyennes annuelles.
Liste des étudiants avant le tri :

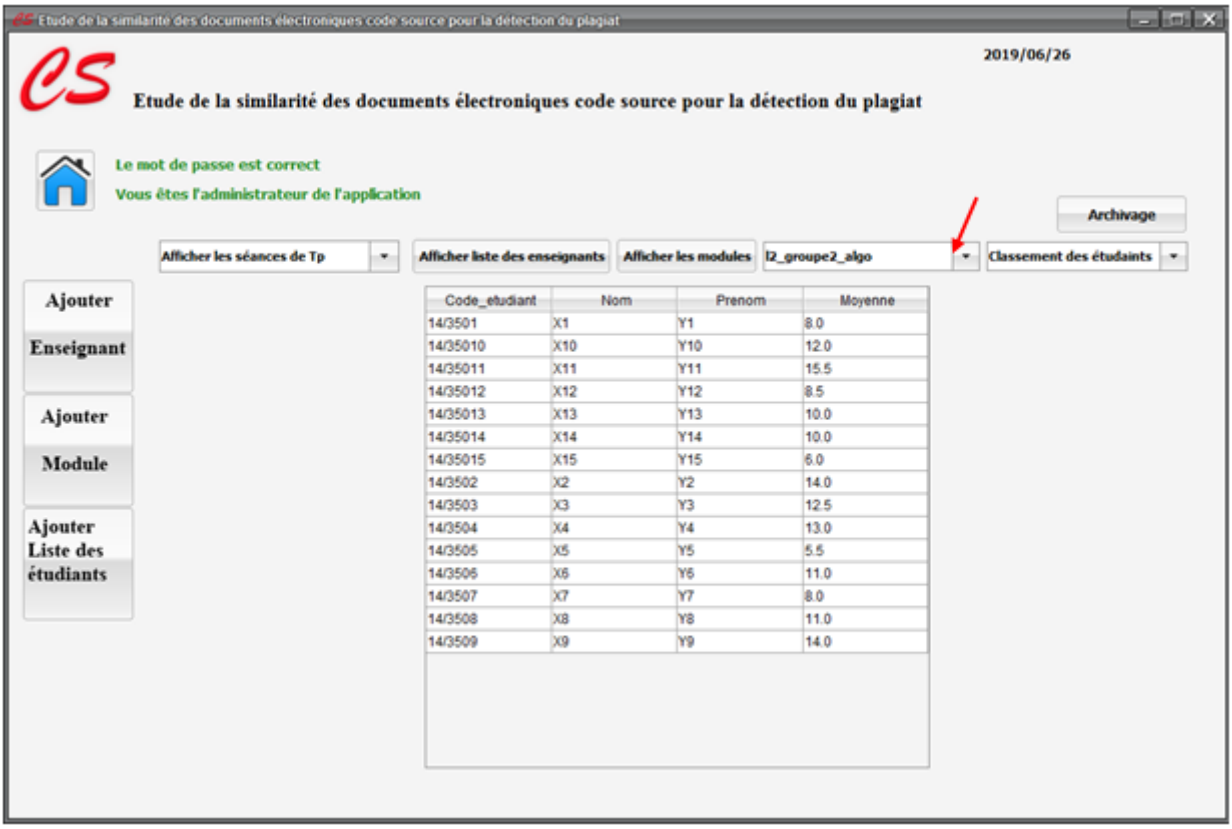


FIGURE 5.12 – Fenêtre d'une liste des étudiants avant le tri

Liste des étudiants après le tri :

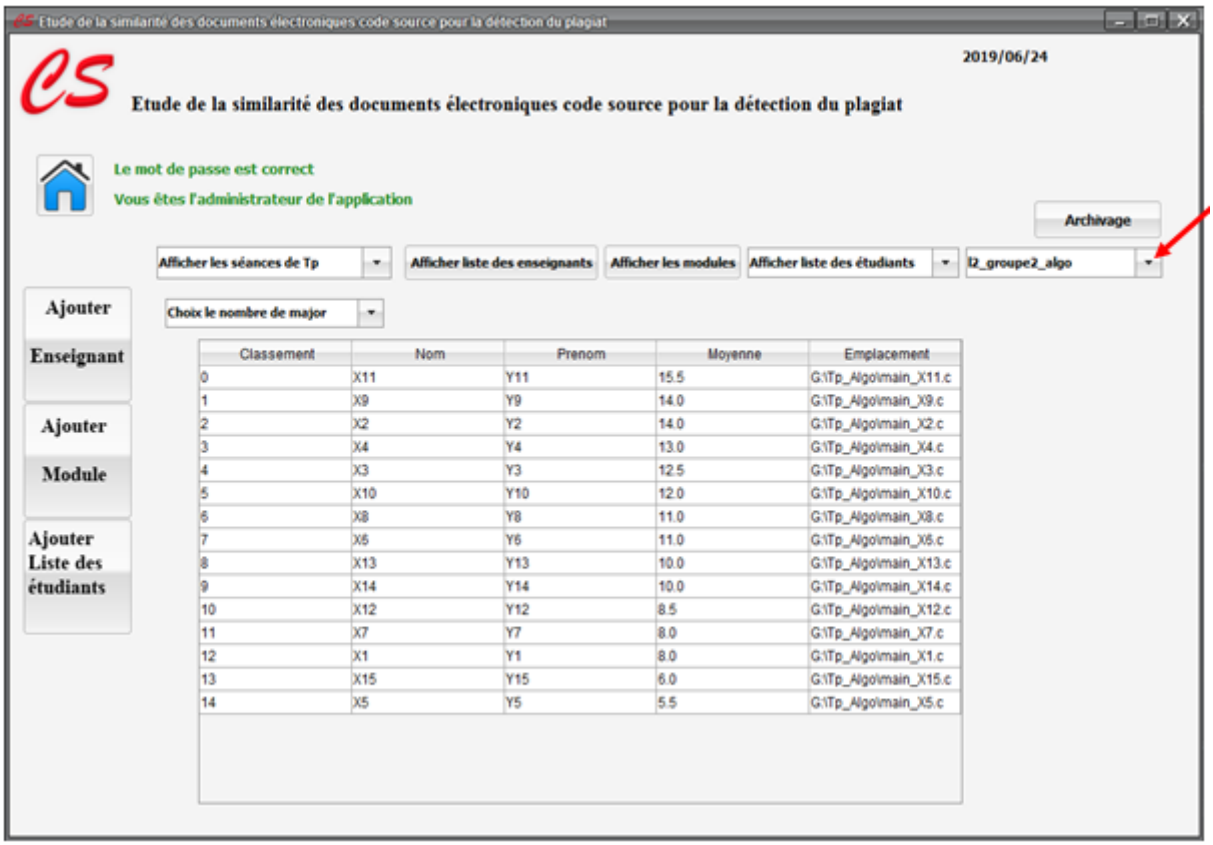
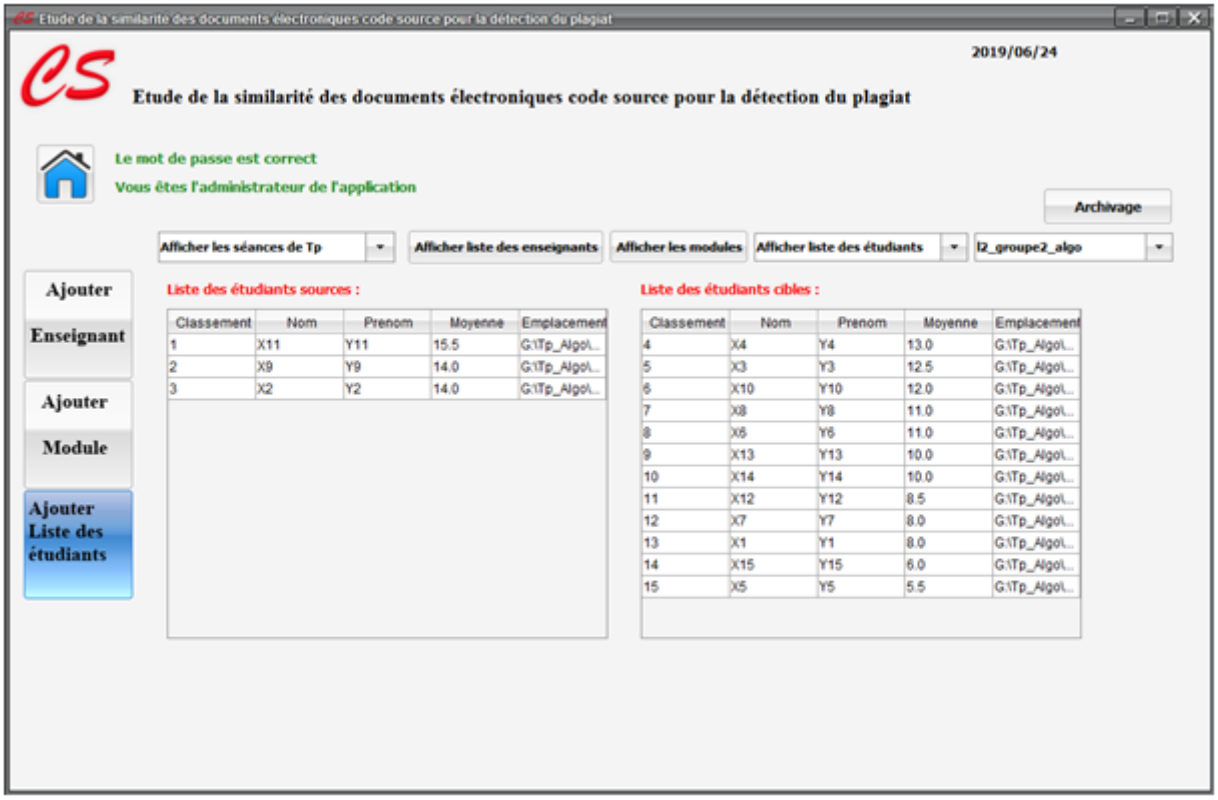


FIGURE 5.13 – Fenêtre d'une liste des étudiants après le tri

Après avoir choisi le tri des étudiants, il fait apparaître un zone de choix de nombre des étudiants majors dans le groupe cette nombre entre 0 et 5 pour séparer les codes sources et les codes cibles.

- Les codes sources sont les codes des étudiants avec des moyennes supérieurs ou égaux le nombre de major choisi serra stocker dans un dossier pour utiliser après dans l'analyse de la similarité.
- Les codes cibles sont les codes des étudiants avec des moyennes inférieurs au nombre de major choisi serra stocker dans un dossier pour utiliser après dans l'analyse de la similarité.



Classement	Nom	Prenom	Moyenne	Emplacement
1	X11	Y11	15.5	G:\Tp_Algo\...
2	X9	Y9	14.0	G:\Tp_Algo\...
3	X2	Y2	14.0	G:\Tp_Algo\...

Classement	Nom	Prenom	Moyenne	Emplacement
4	X4	Y4	13.0	G:\Tp_Algo\...
5	X3	Y3	12.5	G:\Tp_Algo\...
6	X10	Y10	12.0	G:\Tp_Algo\...
7	X8	Y8	11.0	G:\Tp_Algo\...
8	X6	Y6	11.0	G:\Tp_Algo\...
9	X13	Y13	10.0	G:\Tp_Algo\...
10	X14	Y14	10.0	G:\Tp_Algo\...
11	X12	Y12	8.5	G:\Tp_Algo\...
12	X7	Y7	8.0	G:\Tp_Algo\...
13	X1	Y1	8.0	G:\Tp_Algo\...
14	X15	Y15	6.0	G:\Tp_Algo\...
15	X5	Y5	5.5	G:\Tp_Algo\...

FIGURE 5.14 – Fenêtre de choix nombre de major dans un groupe

L'administrateur de notre système a les possibilités de voir et suivre toutes les opérations, les entrées et les sorties précédentes de lui-même ou un enseignant, c'est-à-dire un archivage comme le fiche ci-dessous.

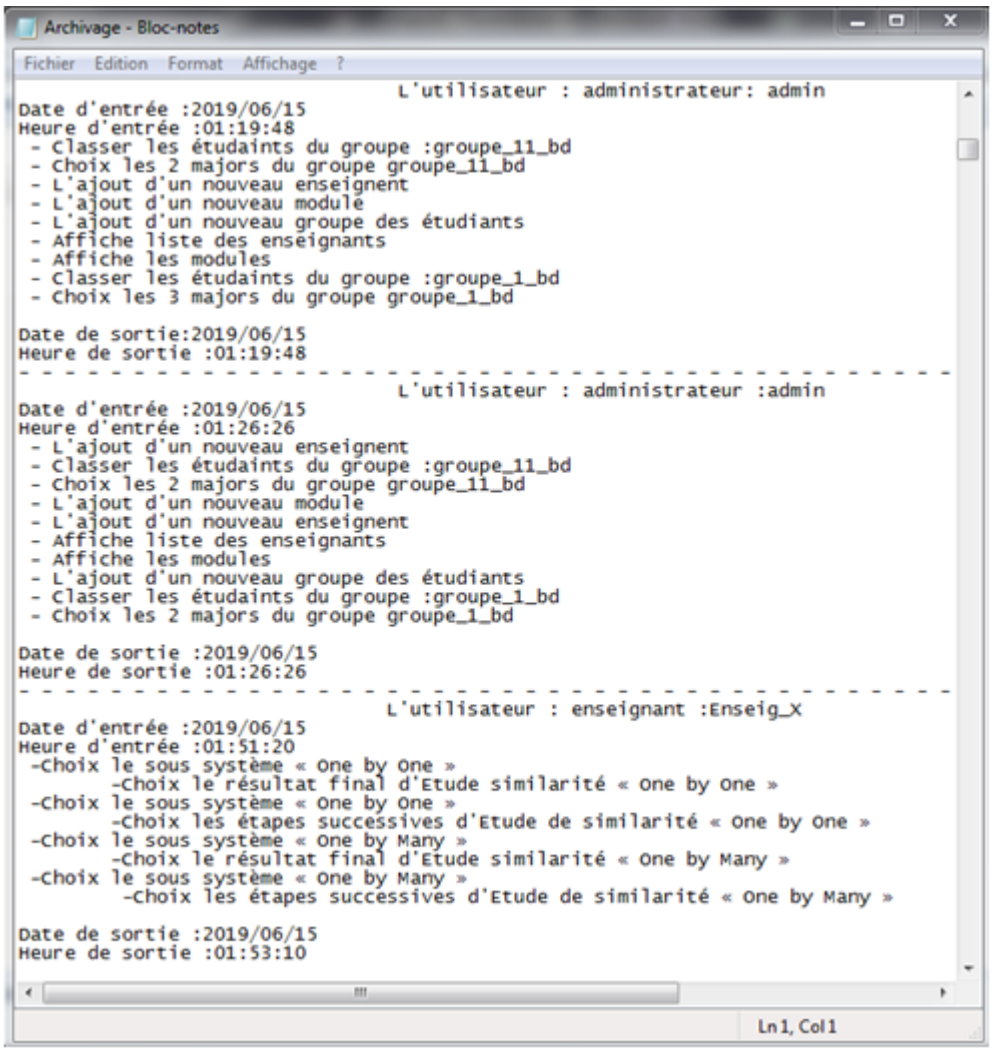


FIGURE 5.15 – fiche d'archivage

5.3.3 Enseignant

L'enseignant est l'utilisateur de système qui a deux fonctions principale qui sont l'exécution de l'analyse de la similarité et l'évaluation des étudiants.
Après l'enseignant choisit d'entrer dans l'application, une fenêtre d'identification va apparaitre pour assure la sécurité de notre application.

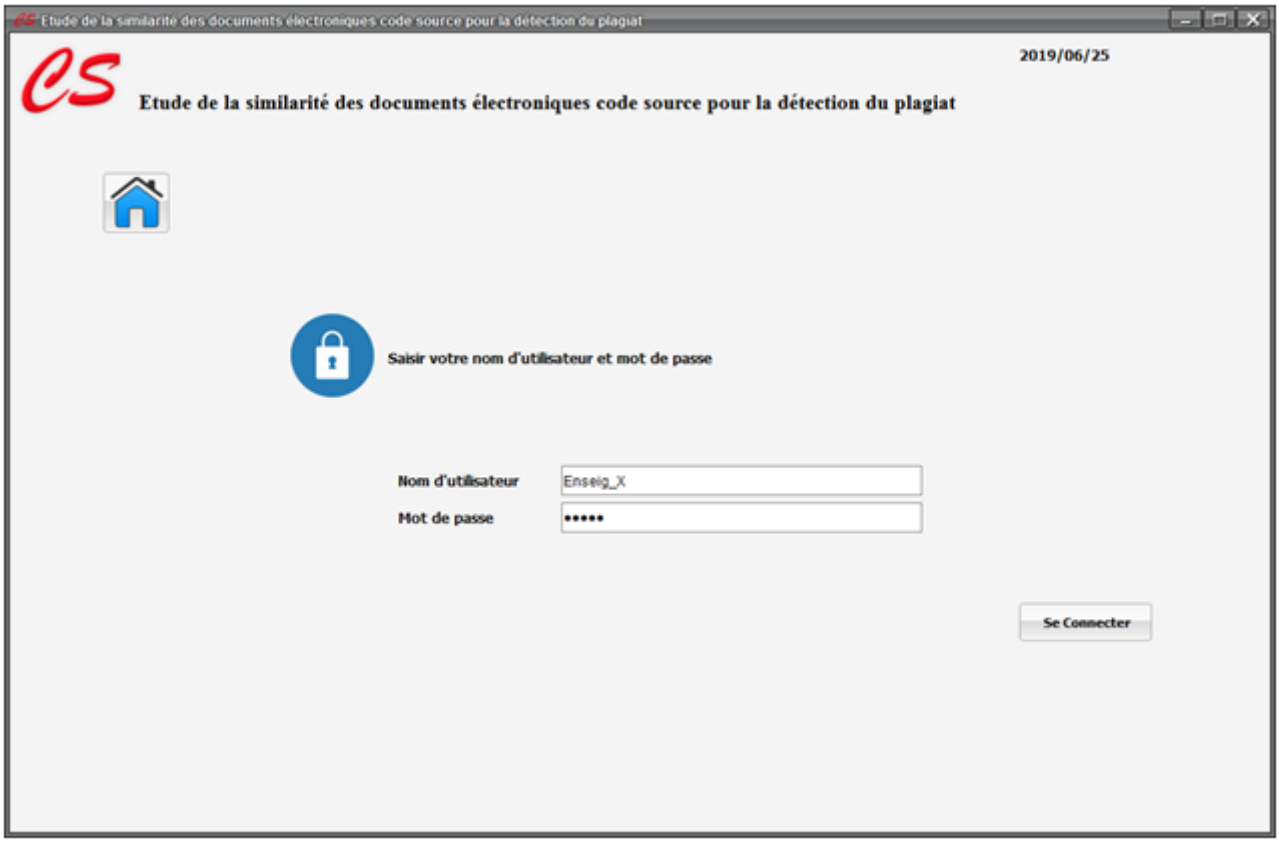


FIGURE 5.16 – Fenêtre d'identification d'un enseignant

la fenêtre de l'enseignant donne la possibilité de :

1. Consulter l'historique.
2. appliquer le sous système « One by One ».
3. appliquer le sous système « One by Many ».

Consulter l'historique :

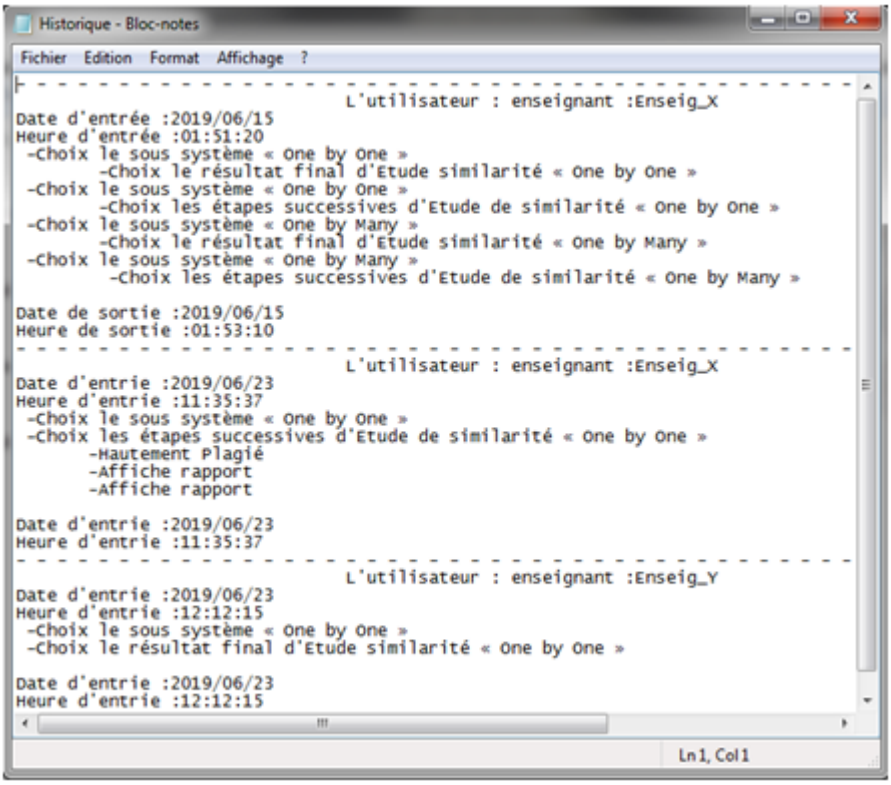


FIGURE 5.17 – Fenêtre d'historique de l'enseignant

le sous système « One by One » :

La figure qui vient représente la fenêtre d'étude de la similarité dans les codes source selon le sous processus « One by One ».
Pour réaliser cette étape on a deux mode de choix :

1. les étapes successive de l'étude de la similarité.
2. le résultat final de l'étude de la similarité.

L'affiche de calcul successive d'étude de la similarité contient des étapes qui sont :

- **Étape 1 :**Sélection.
- **Étape 2 :**Segmentation de Structure.
- **Étape 3 :**Pré-filtrage et substitution et Segmentation.
- **Étape 4 :**Analyse Similarité.
- **Étape 1 :**Sélection.
  La figure suivante montre la fenêtre d'étape de sélection d'un code source et code cible dans langage C.

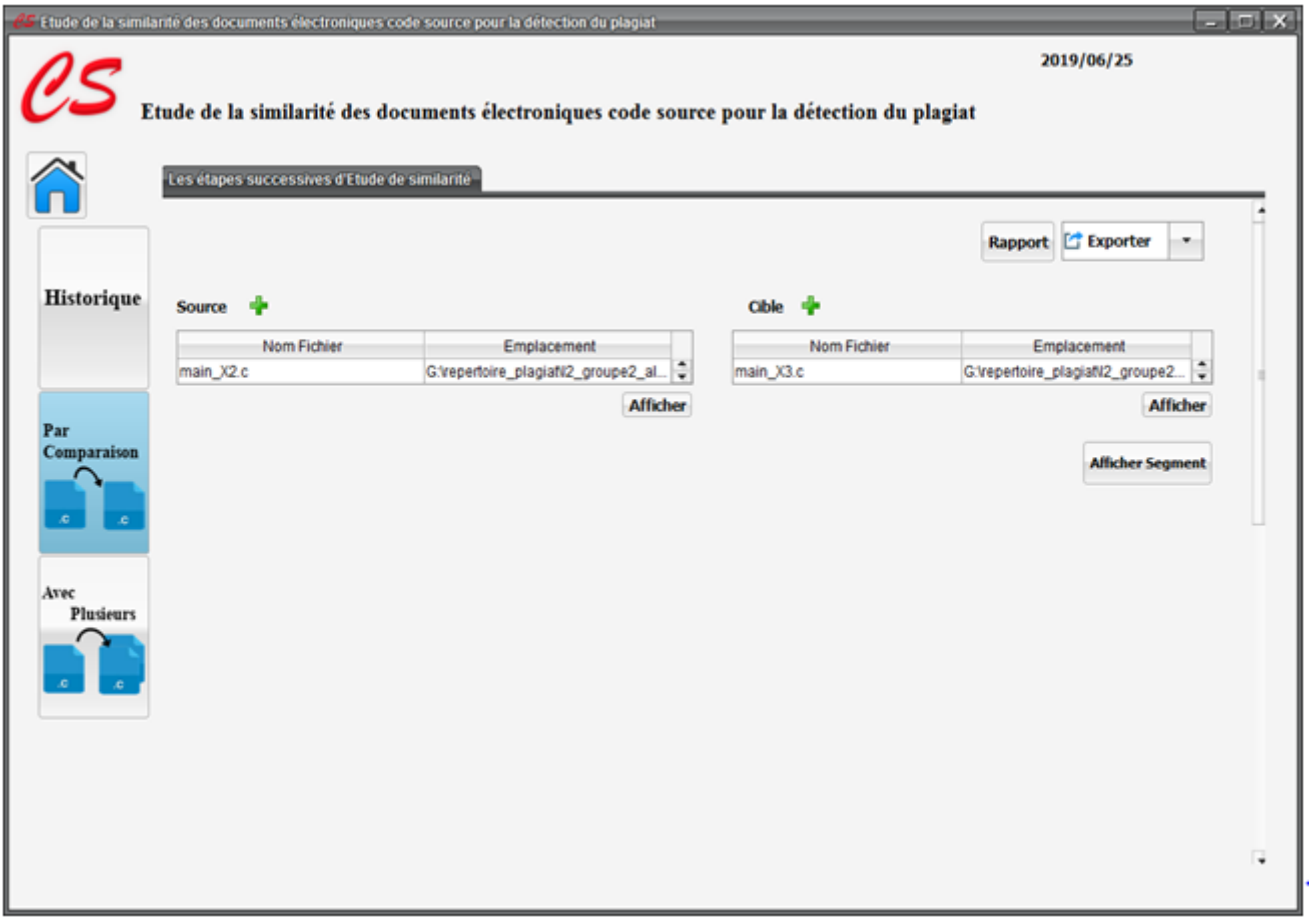


FIGURE 5.18 – Fenêtre de l'étape de sélection

- **Étape 2 :**Segmentation de structure. La figure qui vient montre la fenêtre d'étape de segmentation de structure.

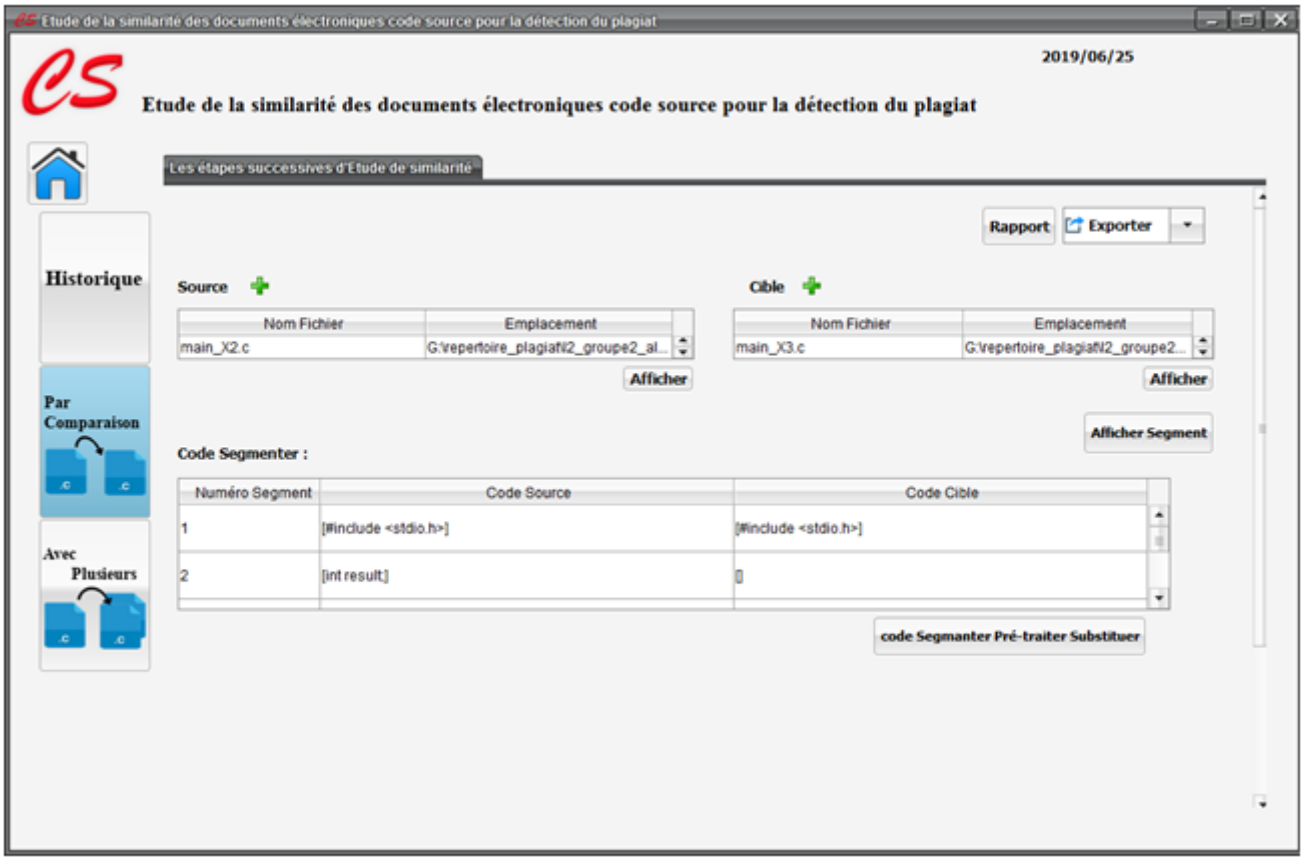


FIGURE 5.19 – Fenêtre de l'étape de segmentation de structure

- **Étape 3 :**Pré-filtrage et substitution et Segmentation. Cette figure suivante représente la fenêtre de l'étape de Pré-filtrage et substitution et segmentation.

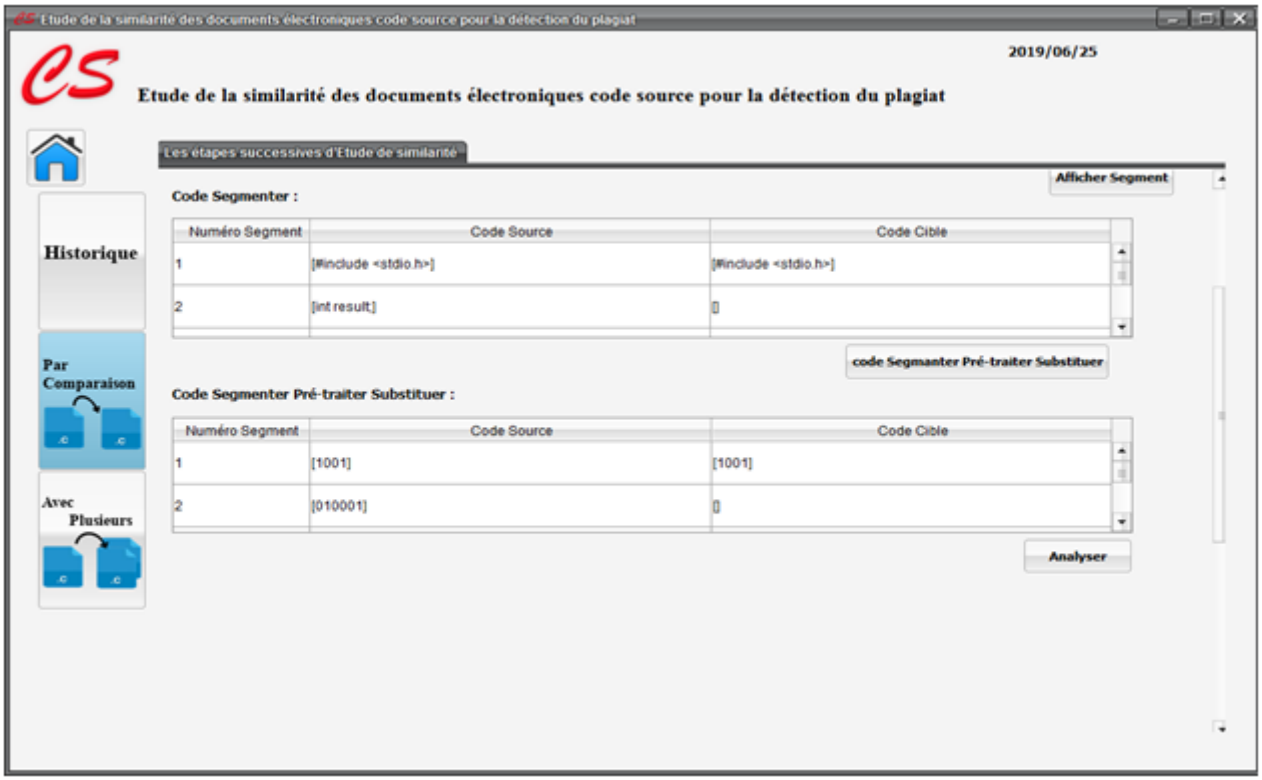


FIGURE 5.20 – Fenêtre de l'étape de pré-filtrage et substitution et Segmentation

- **Étape 4 :**Analyse Similarité. Cette figure suivante représente la fenêtre de l'analyse de la similarité.

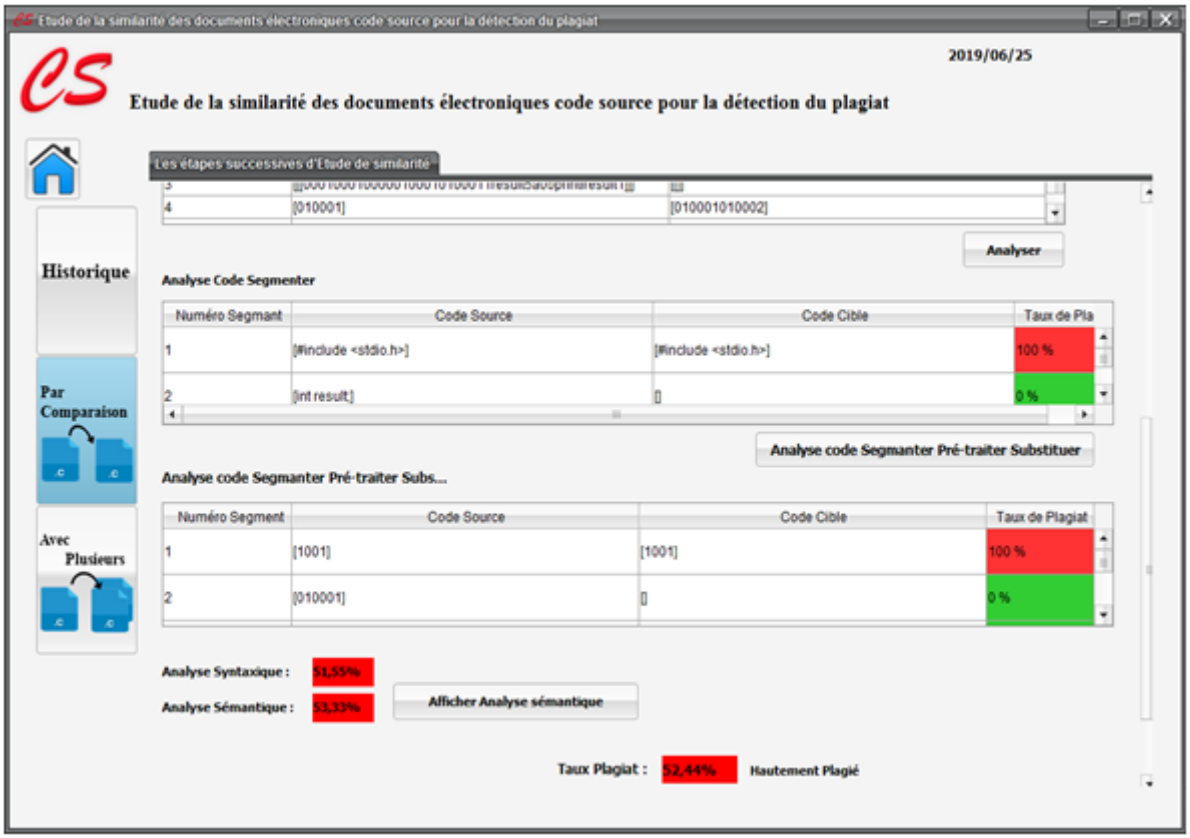


FIGURE 5.21 – Fenêtre de l'étape d'analyse Similarité

L'analyse de la similarité sémantique présente dans la figure suivante.

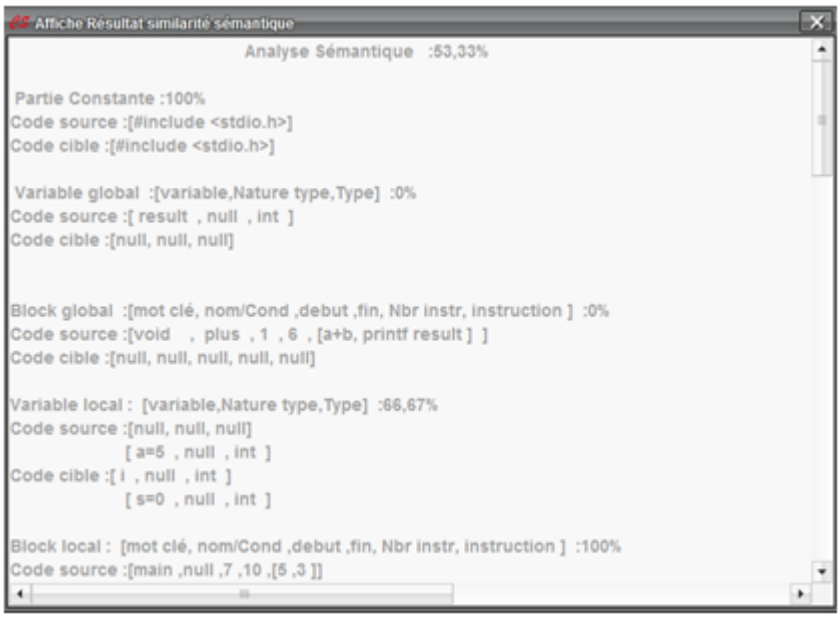


FIGURE 5.22 – Fenêtre de l'étape d'analyse sémantique

On utilise dans notre application un rapport sous forme d'un diagramme circulaire pour faire une représentation visuelle simplifiée et éclaircir le résultat de l'analyse de la similarité.

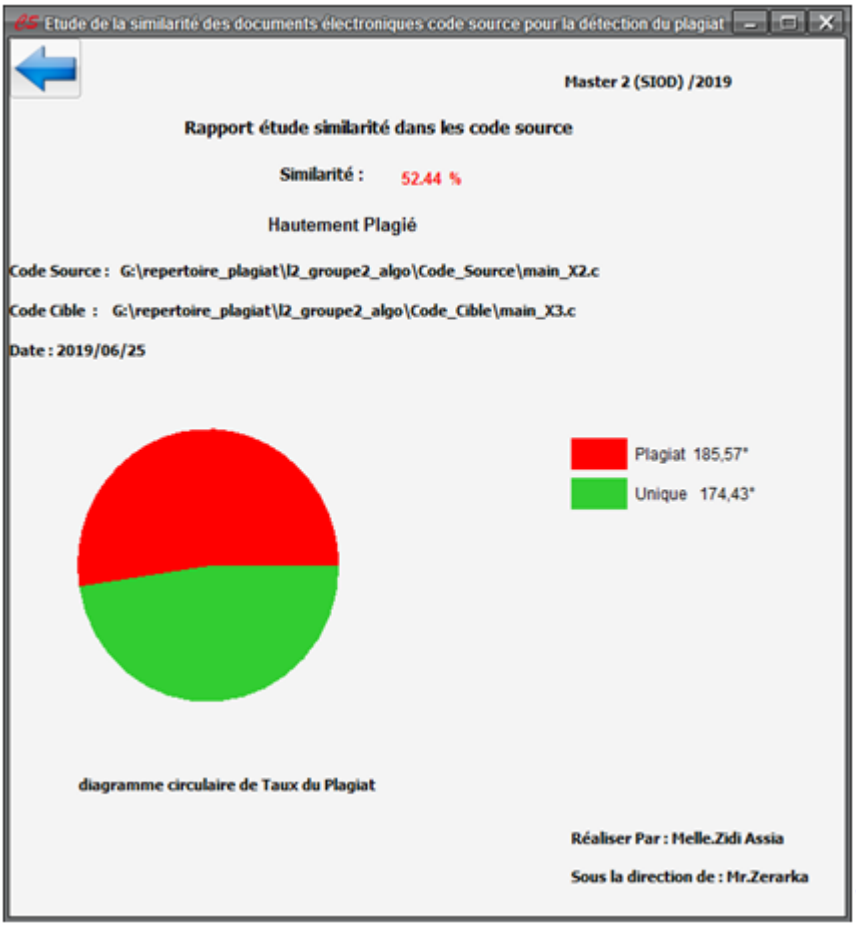


FIGURE 5.23 – Fenêtre de rapport d'analyse sémantique

le sous système « One by Many » :

La section des Figures suivantes montre les étapes de l'étude de l'analyse de la similarité dans le sous système « One by Many ». Pour réaliser cette étape on a deux mode de choix :

1. les étapes successive de l'étude de la similarité.
2. le résultat final de l'étude de la similarité.

ici on choisit le résultat final de l'étude de la similarité.

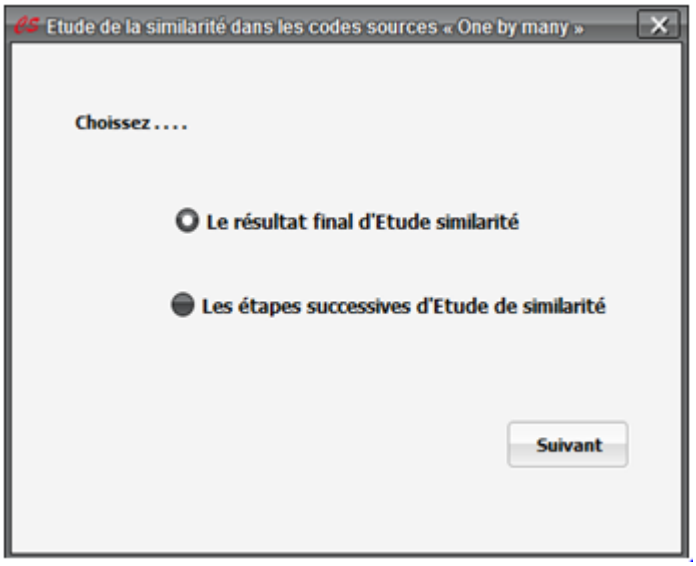


FIGURE 5.24 – Fenêtre de choix le mode de l'analyse « One by Many »

la fenêtre ce-dessous présente le résultat final de l'étude de la similarité de sous système « One by Many ».

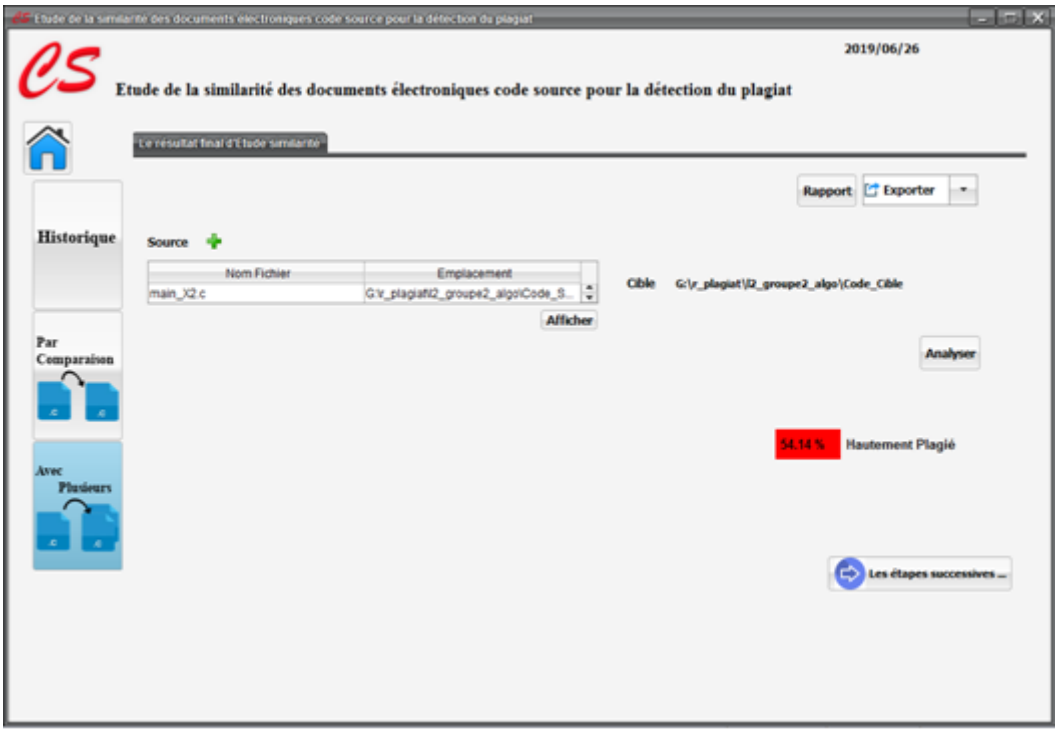


FIGURE 5.25 – Fenêtre le résultat final de l'étude de la similarité de sous système « One by Many »

Cette fenêtre a un bouton qui permet de donner la possibilité de voir les étapes successives de l'analyse de la similarité.
Cette fenêtre présente l'étape de la segmentation de structure et l'étape de Pré-filtrage et substitution et Segmentation.

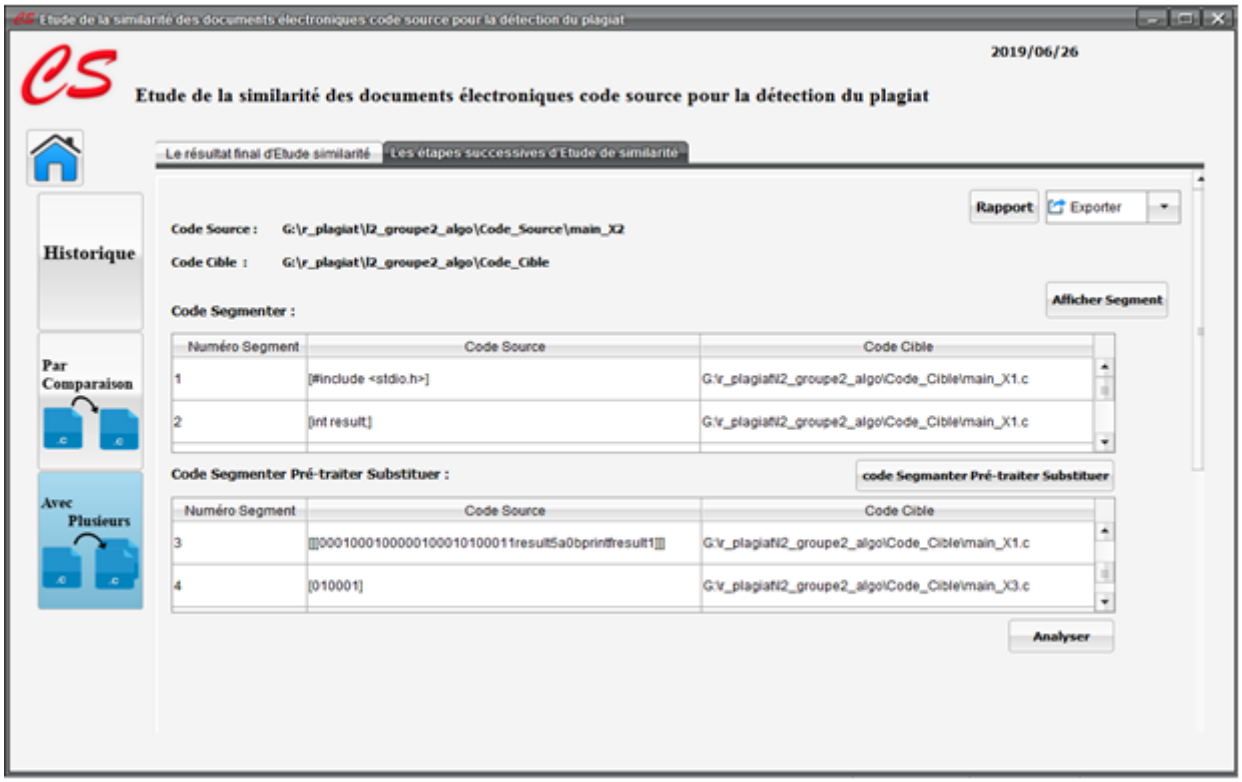


FIGURE 5.26 – Fenêtre segmentation « One by Many »

Cette fenêtre présente l'étape de l'analyse de la similarité.

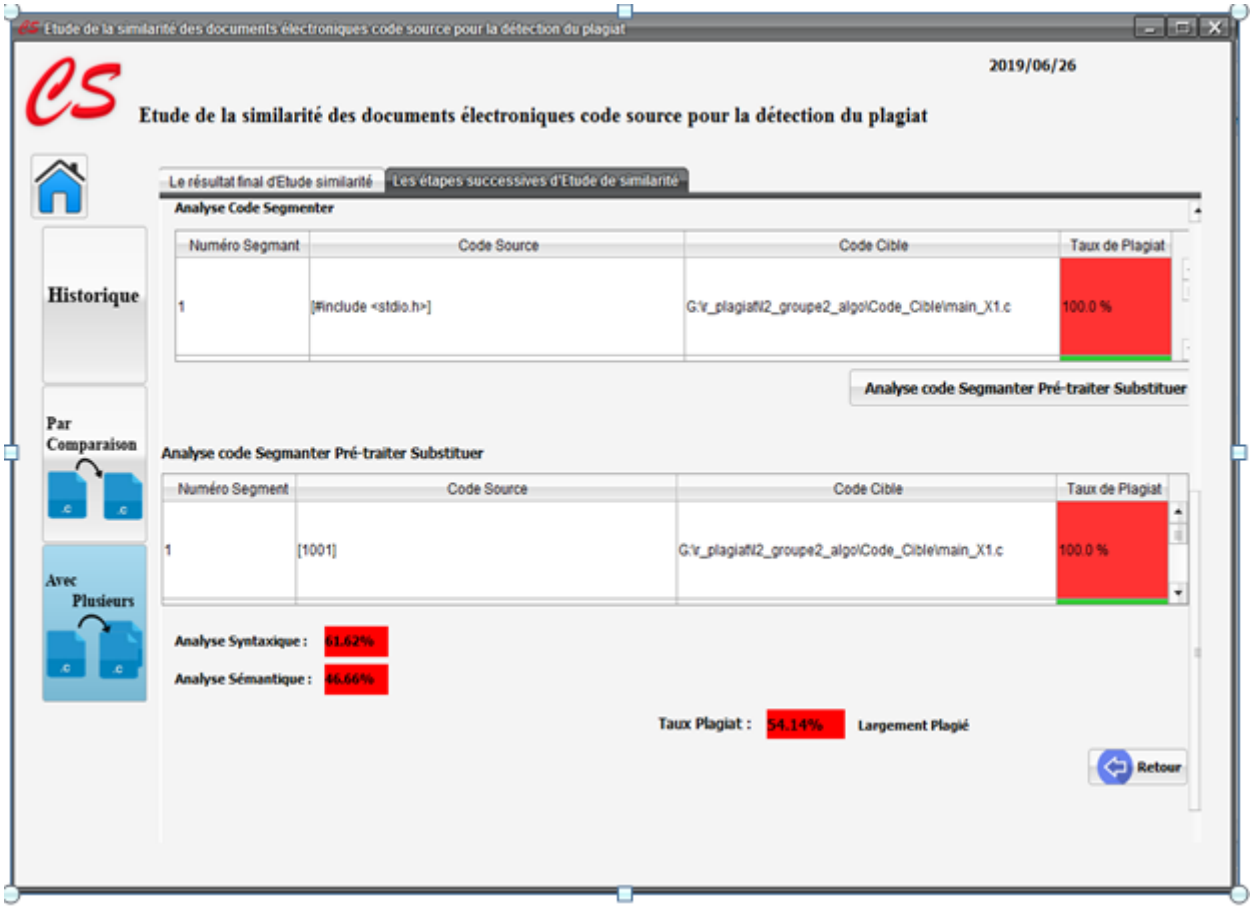


FIGURE 5.27 – Fenêtre l'analyse de la similarité « One by Many »

la fenêtre qui vient représenter le rapport de notre résultat de l'étude de la similarité entre un code source et un ensemble des codes cibles.

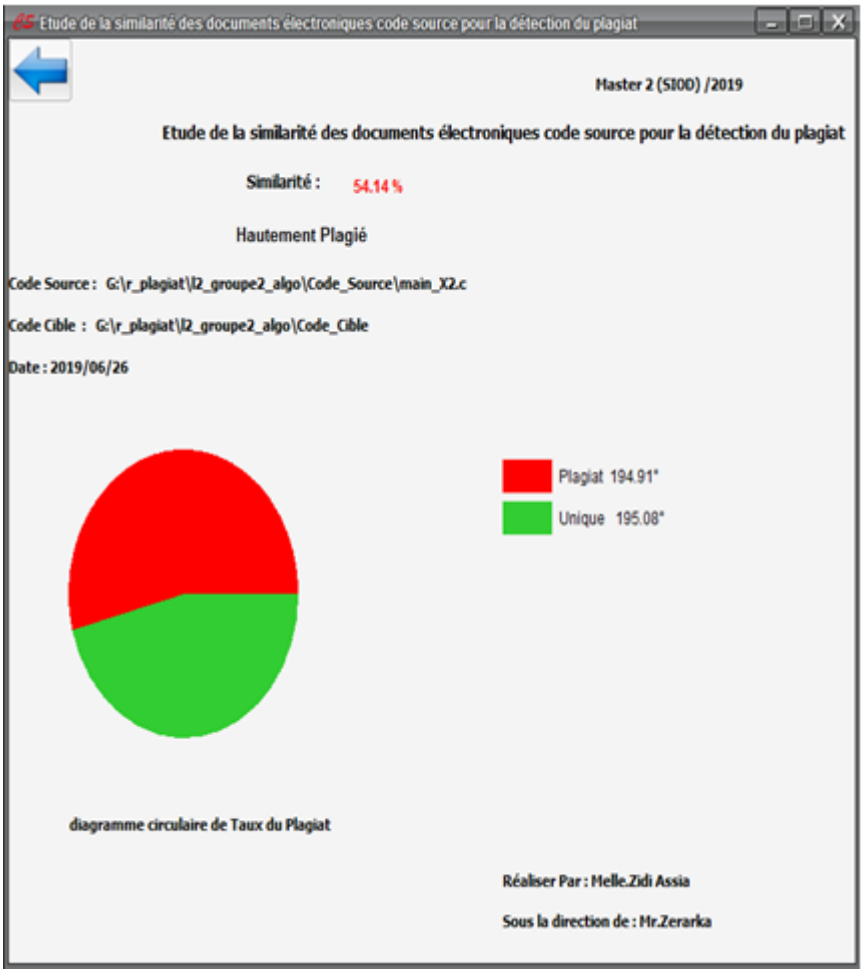
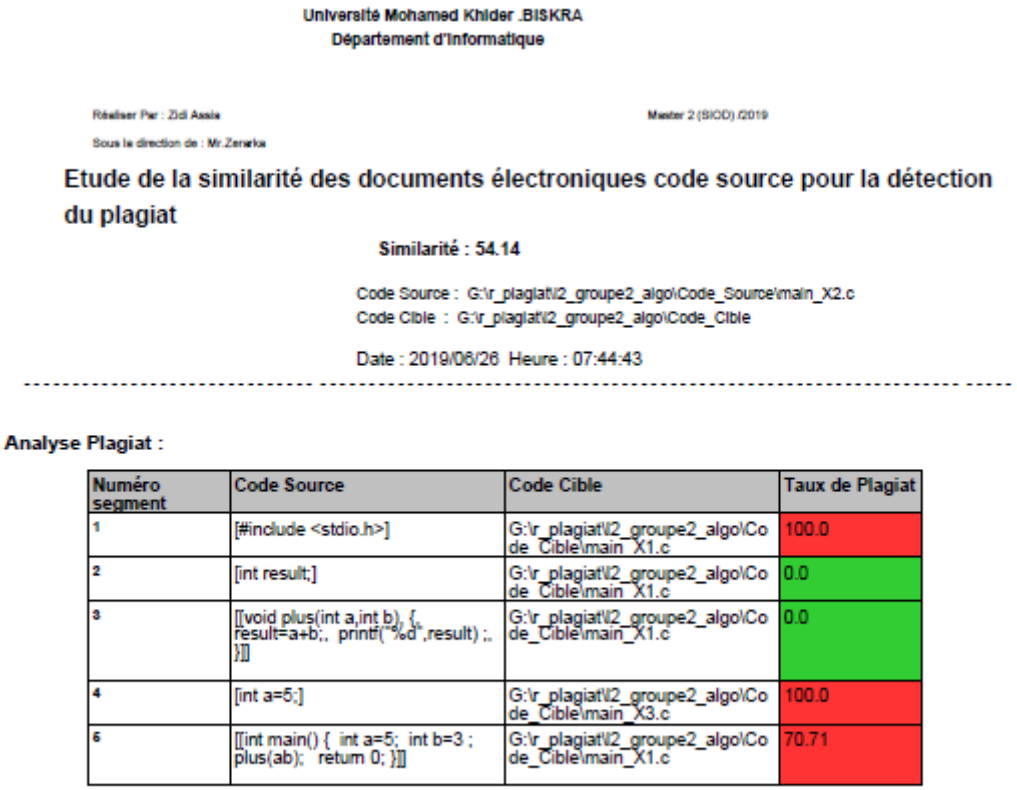


FIGURE 5.28 – Fenêtre de rapport « One by Many »

la figure qui vient représenter le fichier exporté Pdf de notre résultat.



Université Mohamed Khider .BISKRA
Département d'Informatique

Réaliser Par : Zidi Assia
Master 2 (SIOD) /2019
Sous la direction de : Mr.Zeraka

Etude de la similarité des documents électroniques code source pour la détection du plagiat

Similarité : 54.14

Code Source : G:\r_plagiat\l2_groupe2_algo\Code_Source\main_X2.c
Code Cible : G:\r_plagiat\l2_groupe2_algo\Code_Cible

Date : 2019/06/26 Heure : 07:44:43

Analyse Plagiat :

Numéro segment	Code Source	Code Cible	Taux de Plagiat
1	[#include <stdio.h>]	G:\r_plagiat\l2_groupe2_algo\Code_Cible\main_X1.c	100.0
2	[int result;]	G:\r_plagiat\l2_groupe2_algo\Code_Cible\main_X1.c	0.0
3	[[void plus(int a,int b) { result=a+b;, printf("%d",result) ;, }]]	G:\r_plagiat\l2_groupe2_algo\Code_Cible\main_X1.c	0.0
4	[int a=5;]	G:\r_plagiat\l2_groupe2_algo\Code_Cible\main_X3.c	100.0
5	[[int main() { int a=5; int b=3 ; plus(ab); return 0; }]]	G:\r_plagiat\l2_groupe2_algo\Code_Cible\main_X1.c	70.71

FIGURE 5.29 – Fichier exporté Pdf « One by Many »

5.4 Conclusion

Dans ce chapitre, nous avons décrit l'aspect pratique de notre projet. Tout d'abord, nous avons listé l'ensemble des moyens technologiques utilisés (logiciels). Puis, nous avons présenté les différentes interfaces de notre application. A travers cette réalisation, nous avons pu atteindre les objectifs fixés lors de la phase de conception. Lors de développement, nous avons essayé de fournir un ensemble d'interfaces intuitives et simples à utiliser.

Conclusion générale

Tout au long de notre travail, nous avons focalisé tout les efforts à la satisfaction de l'objectif prévue par notre problématique à savoir l'élaboration et la réalisation d'un système d'aide à la décision émanant d'un enseignant quant à l'évaluation les travaux pratiques des étudiants en terme de redondance et de plagiat.

Ces systèmes d'aide n'est au faite qu'une élaboration et réalisation d'un système d'analyse de la similarité des documents électroniques de type code source en langage C. Notre système est composé de deux sous systèmes notamment le sous système « One by One » et le sous système « One by Many ». Il est utilisé par deux acteurs : un administrateur et un enseignant. Ces deux composants de notre système s'appuient sur un ou plusieurs documents électroniques code source ciblé, segmenté et substitué, prétraité dans le but de générer un descripteur syntaxique et sémantique. L'étude de la similarité est élaborée par une analyse syntaxique suivie d'une analyse sémantique. L'analyse est basée sur les mesures des distances notamment celle du modèle vectoriel et la méthode du Cosinus. Quant à l'analyse sémantique par l'évaluation de la robustesse des séquences structurelles des codes intermédiaires générés.

La réalisation de notre système a nécessité l'implémentation d'un environnement Java pour l'application et Xampp Server qui permet la manipulation des bases de données relationnelle (base de données étudiants, base de données enseignant).

Les résultats de la réalisation de notre système d'étude de la similarité des documents électronique de type code source en langage C ont montré la terminaison du système et leur convergence aux données prévue par les objectifs préalablement.

Perspective

Dans le futur, nous envisageons poursuivre ce travail par : la généralisation de notre système pour les autres langages de programmation chacun dans son niveau et sa classe.

Bibliographie

[1] Algorithmique et programmation. https ://www.exoco-lmd.com/fondements-de-lle : 2018-10-23.

[2] Boucles imbriquées. https ://abs.traduc.org/abs-5.3-fr/ch10s02.html. le : 2018-10-30.

[3] Brève histoire des langages de programmation. http ://projet.eu.org/pedago/sin/ICN/1ere/4-langages.pdf. le : 2018-11-03.

[4] Code source. http ://www.dicodunet.com/definitions/developpement/code-source.html. le : 2018-12-20.

[5] Code source : définition, traduction. https ://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203623-code-source-definition-traduction/. le : 2018-12-20.

[6] Data mining/mlapprentissage non-supervisé. https ://www.lamsade.dauphine.fr/ atif/lib/exe/fetch.php ?dia=teaching :coursm2_1.pdf. le : 2019-01-25.

[7] Dictionnaire de français larousse. https ://www.larousse.fr/dictionnaires/francais. le : 2019-01-05.

[8] Droit d'auteur et plagiat. http ://scdautoformation.univ-lyon2.fr/formdoc/SourcesPlagiat/co/Partie2_1.html. le : 2019-03-21.

[9] Du document aux documents. https ://profdoc.iddocs.fr/spip.php ?article18. le : 2018-10-21.

[10] Formats des documents électroniques pris en charge. https ://helpcenter.onlyoffice.com/fr/onlyoffice-editors/onlyoffice-document-editor/helpfulhints/supportedformats.aspx. le : 2018-10-21.

[11] Initiation à l'algorithmique. http ://dbenmerzoug.e-monsite.com/medias/files/algo-chap3.pdf. le : 2018-11-05.

[12] instruction de branchement. http ://www.prism-astro.com/fr/aide/SCRIPTS/Manuel/branchements.html. le : 2018-12-30.

[13] Introduction à "normes et documents numériques : quels changements". http ://gabriel.gallezot.free.fr/Solaris/d06/6introduction.html. le : 2018-11-10.

[14] La documentation française la communication des documents administratifs des collectivités territoriales et de leurs établissements publics. https ://www.ladocumentationfrancaise.fr/var/storage/rapports-publics/004000003.pdf. le : 2018-12-10.

[15] Le dictionnaire informatique. https ://cours-informatique-gratuit.fr/dictionnaire/code-source/. le : 2018-12-20.

[16] Le document : définition et fonctions. http ://www1.univ-ag.fr/buag/cours/LS5-web/co/Cours1.html. le : 2018-10-21.

[17] Le document définition et fonctions. http ://malavoi3.martinique.univ-ag.fr/buag/cours/LS3-web/co/Cours1.html. le : 2018-10-21.

[18] Les 5 types de plagiat - comment les eviter dans votre document ? https ://www.scribbr.fr/le-plagiat/types-de-plagiat/. le : 2019-03-21.

[19] Les instructions de controle. http ://www.iro.umontreal.ca/ pift1969/H04/partie4.doc. le : 2018-11-05.

[20] mise en coherence des dispositions legislatives relatives a l'informatique et aux libertes, a l'acces aux documents administratifs et aux archives. https ://www.senat.fr/rap/l98-248/l98-2484.html. le : 2018-10-12.

[21] Qu'est-ce qu'un document administratif ? http ://www.ville-boissy-saint-leger.fr/Service-Public/voir/F14061. le : 2018-12-10.

[22] Qu'est-ce qu'un langage de programmation. https ://www.01net.com/actualites/quest-ce-quun-langage-de-programmation-502510.html. le : 2018-12-20.

[23] Structure du document. https ://bdd.librecours.net/module/lap4/docUC13.xhtml. le : 2018-10-21.

[24] Xampp : plateforme pour héberger son propre site web. https ://desgeeksetdeslettres.com/web/xampp-plateforme-pour-heberger-son-propre-site-web. le : 2019-06-22.

[25] Xml l'extensible markup language . https ://stph.scenari-community.org/doc/xml/co/docUC13.html. le : 2018-11-21.

[26] *Le langage C Introduction guide de référence*. 2005, 1996.

[27] *Le robert Dictionnaire de français*. 2011, 2005.

[28] *LA COMMUNICATION DES DOCUMENTS ADMINISTRATIFS*, décembre 2017.

[29] *ACCÉDER AUX DOCUMENTS ADMINISTRATIFS*, juillet 2018.

[30] Sylvie Alayrangues, Samuel Peltier, and Laurent Signac. Informatique débranchée : construire sa pensée informatique sans ordinateur. Technical report, Université de Poitiers, Juin 2017.

[31] S. BENTORKI. Analyse de la modélisation de l'entreprise par les processus métier. Master's thesis, Université Mohamed khider, Biskra, 2016.

[32] Gaël Le Boulch. Approche systémique de la proximité : définitions etdiscussion. Technical report, Université Paris IX Dauphine, 2001.

[33] Romain Brixtel, Boris Lesner, Guillaume Bagan, and Cyril Bazin. De la mesure de similarité de codes sources vers la détection de plagiat : le "pomp-o-mètre". Technical report, Université de Caen Basse-Normandie - GREYC, 2009.

[34] Michel CHILOWICZ. *Recherche de similarité dans du code source*. PhD thesis, École doctorale MSTIC Laboratoire d'Informatique Gaspard-Monge (LIGM)l'Université Paris-Est, 2010.

[35] STÉPHANE COUTURE. *LE CODE SOURCE INFORMATIQUE COMME ARTEFACT DANS LES RECONFIGURATIONS D'INTERNET*. PhD thesis, UNIVERSITÉ DU QUÉBEC À MONTRÉAL ET TÉLÉCOM PARISTECH, 1998.

[36] Véronique EGLIN, Stéphane BRES, and Hubert EMPTOZ. Structuration de documents par repérage de zones d'intérêt. Technical report, Laboratoire de Reconnaissance de Formes et Vision RFV INSA de Lyon, 1998.

[37] Sylvain Falardeau and Ghislaine Jetté and. Miieux gérrerr vos documentts admiiniisttrrattiiffs. Technical report, Centre de documentation sur l'éducation des adultes et la condition féminine (CDÉACF), 2014.

[38] Bernard FALLERY and Florence RODHAIN. Quatre approches pour l'analyse de données textuelles : lexicale, linguistique, cognitive, thématique. Technical report, Université Montpellier 2, 2007.

[39] Jean-Philippe Fauconnier, Laurent Sorin, Mouna Kamel, Mustapha Mojahid, and Nathalie Aussenac-Gilles. Détection automatique de la structure organisationnelle de documents à partir de marqueurs visuels et lexicaux. Technical report, Université Paul Sabatier, July 2014.

[40] Jeremy Ferrero. *Similarités Textuelles Sémantiques Translingues : vers la Détection Automatique du Plagiat par Traduction*. PhD thesis, université Grenoble Alpes, 2016.

[41] Nicolas Foucault, Sophie Rosset, and Gilles Adda. Pré-segmentation de pages web et sélection de documents pertinents en questions-réponses. Technical report, Université de Paris-Sud, 2013.

[42] Franck Fourel. Modélisation, indexation et recherche de documents structurés. Technical report, Université Joseph Fourier - Grenoble 1, 5 Février 1998.

[43] OUSSAMA HACHEMANE. Évaluation de l'impact du refactoring basÉ sur les clones sur la qualitÉ (maintenabilltÉ-testabilltÉ) des systÈmes orientÉs objet. Technical report, L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES, 2015.

[44] Siba Haidar. *Comparaison des documents audiovisuelspar Matrice de Similarité. (Video documents comparison using Similarity Matrix)*. PhD thesis, Paul Sabatier University, Toulouse, France, 2005.

[45] Marie-Paule Jacques and Josette Rebeyrolle. Titres et structuration des documents. Technical report, Université Toulouse-Le Mirail, 2006.

[46] Florent Montreuil. Extraction de structures de documents par champs aléatoires conditionnels : application aux traitements des courriers manuscrits. Technical report, UNIVERSITE DE ROUEN U.F.R DES SCIENCES ET TECHNIQUES, 28 juin 2011.

[47] Elsa Negre. Comparaison de textes : quelques approches... Technical report, Laboratoire d'Analyses et Modélisation de Systèmes pour l'Aide à la Décision, 2013.

[48] Mohamed Amine Ouddan and Hassane Essafi. Caractérisation de documents code source basée sur un dictionnaire de grammaire : Application à la détection de plagiats. Technical report, TUNISIA, 2007.

[49] sameh kallel. *gestion et archivage de documents électroniques : évidence, flabilité et authenticité*. PhD thesis, la Faculté des études supérieures de l'université Laval, 1998.

[50] Haïfa ZARGAYOUN. Contexte et sémantique pour une indexation de documents semi-structurés. Technical report, LIMSI/CNRS-UniversitéParis11, 2006.

[51] Xu Zhang. *Analyse de la similarité du code source pour la réutilisation automatique de tests unitaires à l'aide du CBR*. PhD thesis, université Laval Québec, Canada, 2013.