剽窃・コピペの検出とその問題

J10-8014　杢岡　充

1.　はじめに

近年、インターネットの普及や IT の進歩により情報を得るための手段としてこれらが用いられることが多くなった。大量の情報入手が容易になった結果コピー&ペースト（以下コピペ）と呼ばれる行為により容易にネット上から剽窃を行えるようになり、反面その検出は困難となった。

剽窃を未然に防ぐために厳しく剽窃を禁止し罰則を設けたり、リテラシーや情報モラルの教育が一層重視されるようになったりしたが、依然として剽窃行為はなくならない。そのため剽窃を検出するようなシステム、ツールの開発が必要とされている。

金沢工業大学の杉光らによって開発されたコピペ判定支援ソフト（コピペルナー）を導入する大学なども 300 校以上存在している。[1] しかし、コピペルナーによる判定では不十分と判断し他の手法による検出方法も研究されている。

2.　取り扱う剽窃について

剽窃とは他人の作品などを自分のものとして発表することなどである。本論では特に教育の場で発生しやすい課題・レポート内での剽窃を対象とする。またコピペとは本来パソコンのコピー&ペースト機能を用いることであるが、本論では手作業で右から左に書き写すような行為についても対象とする。

3.　研究目的

剽窃による弊害は種々あるが、それらを解決するためにはより精度の高い検出方法が求められる。その検出方法は多岐にわたっているが、課題・レポートの内容や出題形式によって検出方法の向き不向きが存在すると考えられる。そこで既に研究されている手法において、その特徴や対象となった内容を検証していくことで対称と検出手法の相性を明らかにすることを目的とする。

4.　関連研究

4.1.　自然言語を対象としたもの

宮本らはレポートの作成過程に着目し、その文字数増減の様子からコピペを使用した箇所を摘出できると考えた。Google ドキュメントの編集履歴から短時間での不自然な文字数の増加をヒントに検出を試みた。文字数の多いコピペについては検出に一部成功しているが、コピペかどうか判断できないケースも見受けられた。(表 1)

表 1　大幅な増加文字数についてコピペかどうかの判断理由

1.　コピペであると判断した理由
（1）　削除・訂正を伴わない文章が瞬間的に増える ・1 分間に訂正なく 100 字以上の増加 ・編集作業のはじめに 100 字以上の増加 ・非常に不自然に,文章の途中からいきなり完成している ・前の履歴と同時刻で増加文字数を入力する時間はない ・段落としての固まりで増えている （2）　不自然さ ・文体（言い回し）の変化が見られる ・改行が多い・不自然な改行になっている（メールからのコピーの可能性が高い） ・フォントサイズが変わっている
2.　実際に入力されたと判断した理由
（1）　文章を作成している雰囲気 ・履歴を辿ると少しずつ文章作成を進めている ・文章の途中から入力が始まり、途中で終わっている ・入力が次の履歴で続いている （2）　レポートに合った段落構成 ・1 履歴の中で複数の箇所を書いている ・本文に合わせた改行になっている ・箇条書きが数行 （3）　手入力特有の現象 ・入力ミス・変換ミスがある ・文末が入力途中のアルファベットで終わっている ・変換途中の漢字が見られる
3.　コピペかどうか分からないもの
・入力した可能性が高いが判断は難しい ・大幅に文字数は増えているが、入力する時間は十分にある ・前の履歴との時間が大きく空いているため判定できない

深谷らは単なるコピペに限らずそれらに手を加えた行為を「真似」と称し、真似にも対応できる検出手法を提案した。基本となる類似性の測定には単語の頻度

統計を用いるが、その際に類義語や同義語をもちいた単語の置換による真似を想定しあらかじめ類義語などの類似性を定量化することで対応した。

4.2. プログラミング言語を対象としたもの

上田らはレーベンシュタイン距離を用いてソースコードの類似性から剽窃の検出を試みた。レーベンシュタイン距離はソースコードの字面から直接類似性を算出するため、対象となるなる文字列の長さに比例して計算処理にかかる時間が増加してしまう。そこで上田らは字句解析によってソースコードをトークンに区切り文字列に対してではなくトークン列に対して検査した。、

$$LEV(S,T)=\begin{cases} |S| & ;\ T=\phi \\ |T| & ;\ S=\phi \\ LEV(S',T') & ;\ \text{先頭が一致} \\ \min\left\{\begin{array}{l} LEV(S',T) \\ LEV(S,T') \\ LEV(S',T') \end{array}\right\}+1 & ;\ \text{不一致} \end{cases}$$

レーベンシュタイン距離の定義

井岡らはソースコードそのものに対してではなく、プログラムの実行履歴に着目しプログラムの挙動から類似性を求めて剽窃を検出する手法を提案した。実行履歴の中からメソッドの呼び出し、コンストラクタの呼び出しのみに着目し、メソッドの類似性とクラスの類似性を求めた。

5.考察

自然言語においても、プログラミング言語においても文字数の増減という現象は共通点であるので、宮本らの研究のように過程に着目した検出方法はプログラミング言語における検出方法にも応用できると推測される。ただし、自然言語とプログラミング言語では時間あたりに増加する文字数が同じでないことが予想されるため、コピペの検出のためには閾値を吟味しなおす必要がある。

深谷らのように単語に着目した手法は、使用される単語、すなわち字句が自然言語に比べて大幅に限られるプログラミング言語において有用であるだろう。ただし、文字数が制限されるため字句により構成される空間が小さい場合必然的にすべてのコードで類似性が高くなるため、一定量以上の長さを持つソースコードに対しての使用が望ましいと推測する。

井岡らの手法はプログラムの挙動から類似性を求めるため、例えば「繰り返し構文を用いて配列の中身を印字せよ」というような単純なプログラムでは満足のいく結果は得られないと考える。よって、クラスやメソッドを複数用いて一定の独自性を許すような規模のプログラムに対して用いることが適している。

6. おわりに

今回得られた予想をもとに複数の対象に対して複数の手法を用いて検出精度がどのように変わるのか実証していきたい。

また、今回扱わなかった手法として他にもソースコードのネストや構造に着目する手法などもある。それらについても同様に検証していきたい。

参考文献

[1] 株式会社：コピペルナー,株式会社アンク(オンライン), 入手先〈http://www.ank.co.jp/works/products/copypelna〉(参照 2014-07-03)

[2] 宮本淳,仙石昌也,山森孝彦,久留友紀子,橋本貴宏,:Google ドキュメントを利用したレポート作成過程の分析,日本教育工学会研究報告集,13(4),pp1-6,(2013)

[3] 深谷亮,山村毅,工藤博章,松本哲也,竹内義則,大西昇,単語の頻度統計を用いた文章の類似性の定量化-部分的類似性の考慮-,電子情報通信学会論文誌,J87-D-Ⅱ(2),pp661-672,(2004)

[4] 上田和志,富永浩之,:類似性に基づくレポート剽窃の検出ツールの改良とソースコードへの適用,情報処理学会研究報告,コンピュータと教育研究報告,107(9),pp1-8,(2010)

[5] 井岡正和,吉田則裕,井上克郎,プログラム実行履歴を用いた類似クラス・メソッド検出手法,コンピュータソフトウェア科学会コンピュータソフトウェア,31(1),pp110-115,(2014)