4N-04

コピー対策のための編集距離に基づく文書コピー度評価

林　成元† 張　馨雲† 石田　将暉† 辻田　航希† 成　凱†
九州産業大学†

1. はじめに

昨今の高度情報化社会において、膨大な量の情報が誰でも入手可能になっている。確かに、これは社会にとって大きな進歩であるが、その反面、様々な問題も生み出している。その一つとして、レポート等における“コピー文書“が挙げられ、コピー対策が求められている[1]。我々は、ある文書がコピーであるかを定量的に評価するプログラムを作成するため、基本単位を文字単位、単語単位とし、また、文書を集合としてモデル化したものを「集合距離」、系列としてモデル化したものを「編集距離」によってそれぞれコピー度を計算した。事前に類似度及びパターンの異なるコピー文書を用意し、プログラムによって評価したものが一致するかどうかを検証し、有効性を確かめた。

2. 文書類似度

コピー度を計測する基礎となる文書類似度を紹介する。要素の順序を考慮する場合と考慮しない場合によって類似度の評価方法が違う。

2.1 Levenshtein 類似度

Levenshtein 類似度は編集距離[2]に基づく文字列の類似度である。編集距離とは二つの文字列を 1 文字の挿入・削除・置換によって一方の文字列をもう一方の文字列に変形するのに必要な手順の最小回数として定義され、レーベンシュタイン距離とも呼ばれる。長さがそれぞれm,nの文字列x,yの編集距離を$d_L(x,y)$とするとき、Levenshtein 類似度は次のように定義する。

$$1-\frac{d_L(x,y)}{\max\{m,n\}}$$

2.4 Jaccard 類似度

Jaccard 類似度は，集合間の類似度をはかる尺度として Jaccard index や Jaccard similarity coefficient と呼ばれる。ある集合 A と別の集合 B についての Jaccard 類似度 J(A,B)は，以下の式で定義する。

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|}$$

集合 A と B が空集合 ϕ の時，J(A,B)＝1 とする。

Jaccard 類似度は 2 つの集合の差集合の要素数に大きく依存するため，差集合の要素数が多いほど Jaccard 類似度は小さくなる。

2.3 Simpson 類似度

Simpson 類似度は集合間の類似度をはかる尺度として Overlap coefficient や Szymkiewicz-Simpson coefficient と呼ばれ、集合 A と集合 B Simpson 類似度は次のような式で表される。

$$\frac{A\cap B}{\min(|A|,|B|)}$$

Simpson 係数では要素数が少ない方の要素数を分母としているため，一方の集合の要素数が少ない場合に，差集合の要素数がどれだけ多くても類似度がほぼ 1 となってしまう。

Jaccard 類似度や Simpson 類似度などの集合の類似度は直接的に文書の類似度を反映できるが、計算公式が定められているため、カスタマイズが困難である。それに加えて、集合の類似度は単語順序を考慮しないので、これをもって文書のコピー度を判断するのは難しい。一方、Levenshtein 距離は単語順序を考慮しているものの、長い文書や構成変化などの状況に対して、文書のコピー度の結果から大幅に外れる可能性が高い。

3. 文書類似度に基づくコピー度評価

文書モデルによってコピー度の評価方法が異なる。まず、**文書の構造を考慮せず単に文字の羅列としてモデル化**することができる。このとき、前述の文書類似度そのままコピー度の評価に適用できる。そして、文書を段落の集まり、段落を文の集まり、文を単語の集まりとして**文書構造を考慮してモデル化**することもできる。

文書構造を考慮したコピー度評価は次のように行う。まずは文閾値と段落閾値を設定する。Levenshtein 距離に基づいて、文単位で比べて編集距離を計算し、文閾値と比べて文単位のコピー度を得られる。次に、文単位のコピー度をもって、同じ方法で段落閾値と比べて計算し、段落コピー度を得られる。なお、一般的に、文は

The Evaluation of Document Similarity Based on Edit Distance for Plagiarism Detection
†Chengyuan Lin, †Xinyun Zhang, †Masaki Ishida, †Koki Tsujita, †Kai Cheng
†Kyushu Sangyo University

形態素解析を使用し名詞と動詞に分類され、分類後の単語単位で保存される。

上記の手法を具体的に説明する。二つの文書D_1と文書D_2のコピー度を求めるとき、それぞれを段落の集合としてD_1={$P_{1,1}$, $P_{1,2}$, …, $P_{1,m}$}とD_2={$P_{2,1}$, $P_{2,2}$, …, $P_{2,n}$}がある。そして、各段落$P_{1,i}$と各段落$P_{2,j}$の下に文の集合として$P_{1,i}$={$S_{1,1}$, $S_{1,2}$, …, $S_{1,p}$}と文$P_{1,j}$={$S_{2,1}$, $S_{2,2}$, …, $S_{2,q}$}がある。まずは文$S_{1,i}$と文$S_{2,j}$の Levenshtein 距離を交互に計算し、文閾値T_sと比べて、文閾値T_sより大きい場合にはC_sを加算する。これにより、段落コピー度P_{sim}=C_s/(m*n)が得られる。次に、段落$P_{1,i}$と段落$P_{1,j}$を交互に計算し段落閾値T_pと比べて、段落閾値T_pより大きい場合、C_pを加算する。これにより、文書コピー度D_{sim}=C_p/(p*q)が得られる。それ以外に、段落閾値を除いて、文閾値(only)だけを考える手法もある。

このように、我々の提案手法では単純な Levenshtein 距離のデメリットを軽減し、集合の類似度により相対的に合理的なコピー度が期待される。

4. 評価実験

提案手法を評価するために、実データによる実験を行った。人間により長さやコピー度の異なる文書を複数用意して、これらの文書を提案のアルゴリズムを使って評価を行う。

4.1 実験データとパラメータ

提案手法の評価のために、2 つのデータセットを用いる。データセット 1 は 800 字程度の文書をベースとして語尾、構成、単語変化などを加える複数の文書である。データセット 2 は 3000 字程度の文書をベースとして一定範囲内のアレンジを加え、おおよそ 20%ごとにコピー度を上昇する複数の文書である。それ以外に、提案手法では文閾値の三つのバリエーションと文閾値(only)と Jaccard 類似度、Simpson 類似度、Levenshtein など、これらの手法で計算したコピー度を比較する。なお、実験は主に文閾値に焦点を当て、段落閾値はデフォルト値(0.1)とする。

4.2 実験結果

実験結果は図 1 に示した。提案手法の実験結果は他に比べると相対的に安定性を保った。語尾、構成、単語などの文字変化は提案手法の実験結果に与える影響が有限である。

図 1 と図 2 から見ると、提案手法は Levenshtein 距離のデメリットを改善することが確認できた。また、集合の類似度の手法を比べると、我々の提案手法ではコピー度に対し、より敏感かつカスタマイズしやすい。これにより、

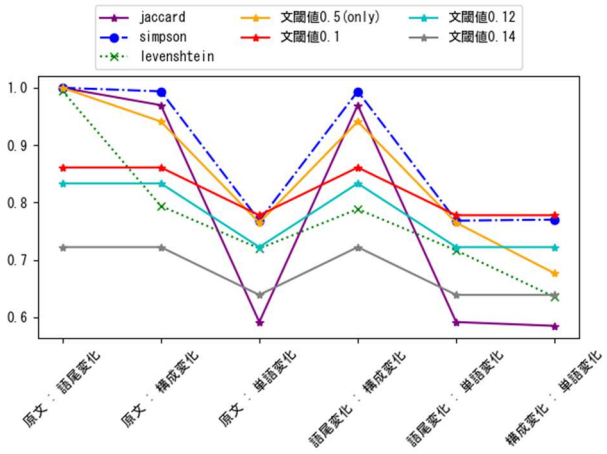

図 2　データセット 1 の実験結果

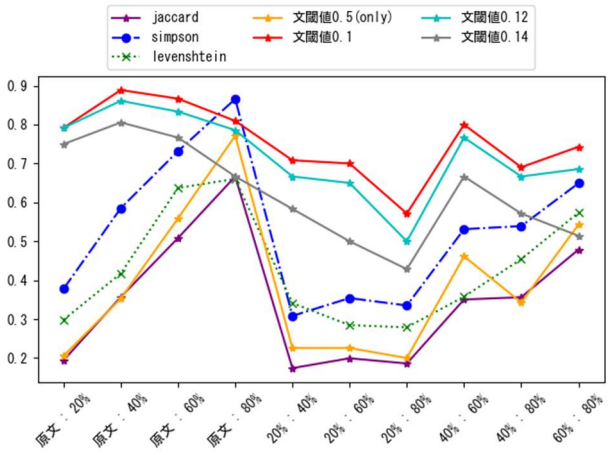

図 1　データセット 2 の実験結果

我々の手法が相対的に優れていることを確認できた。よってある程度実験課題の問題を解決したといえる。

4.3 考察

実験結果の誤差の可能性もあるが、文閾値の調整は単なる縦軸方向の移動ではないと考え得る。今後の課題として、段落閾値の評価や文が、文書での位置によりコピー度における重みを変化させることを反映させることによってコピー度の精度をより上回る可能性がある。

5. 終わりに

本研究では、既存の類似度計算方法を参考し、コピー度の計算方法を提案し、実験を通じて、様々な計算方法と比べて評価した。

参考文献

[1] 杉光一成 (2016)，大学等における「コピペ」問題の現状と対策及びその課題，情報処理学会研究 報告 情報基礎とアクセス技術 (IFAT)， 2016-IFAT-120(6),1-1 (2016-01-18)，2188-8884

[2] Michel Marie Deza, Elena Deza (2016), Distances on Strings and Permutations, pp.215-228, Chapter 11, Encyclopedia of Distances, 4th edition, Springer 2016