模倣レポート判定支援システムの開発

太田 貴久　　増山 繁
豊橋技術科学大学 知識情報工学系
{kikyu, masuyama}@smlab.tutkie.tut.ac.jp

1　はじめに

近年，情報技術の発達により他人のレポートやインターネット上の文章をコピー&ペーストして，それに多少の手直しを加えたレポート (以下，このように作成されたレポートを「模倣レポート」と呼ぶ) を簡単に作成できるようになった．教師は何十，場合によっては何百とあるレポートの中から，このような模倣レポートを判別しなければならないが，この作業は困難を極める．そこで，レポートの模倣の程度を定量化し，それを判定するシステムを作ることが出来れば，教師の仕事量を減らすだけでなく，その存在自体が模倣レポートの抑制に繋がると考えられる．

一言にレポートの模倣と言っても，オリジナルと模倣との間の関係や程度には様々なものが存在する．関係を挙げると，完全な「原稿」が存在するようなオリジナルと模倣が “1 対 1”の関係となるものもあれば，複数の「原稿」から部分的に切り取った文章を繋ぎ合わせた，オリジナルと模倣が “多対 1”の関係となるものもある．また，模倣の程度では，換言のレベルや構文構造の類似度など様々な尺度が存在する．本システムはこのような模倣の関係や程度を考慮に入れ，レポートの類似している部分と，その程度を利用者に提示する．

これまでに模倣レポートの判別を目的とした研究は文献 [1], [2], [3], [4] くらいである．文献 [2] では $tf \cdot idf$ を用いたベクトル空間法による類似度計算法を提案し，文献 [1] では n-gram 解析を用いた類似度計算法を提案している．[2] は文章を単純な単語の集合として取り扱い，その $tf \cdot idf$ 値ベクトルのコサイン尺度で類似度を求める方法であり，一方，[1] は文字の n-gram 頻度解析による方法である．これらの方法では全体的に模倣をしている文章を検出することは出来ても，部分的に模倣している文章を検出することは出来ない．また，文献 [3] では文書を文単位に分割し，各文の同義語／類義語を考慮した単語の頻度ベクトルを用いて文章の類似度を計算する手法を提案している．この方法では，文の結合／分割が発生した場合に本質的に対処ができないという問題点がある．最後に，文献 [4] で太田らは構文情報を用いた文の結合／分割に対応した手法を提案している．ここで，これら全ての手法は文書間の類似度を 1 つの値で評価するので，部分的な模倣を総合的に評価することが出来ないという欠点がある．そこで，本システムでは太田らの手法に改良を加えた方法を用いて文章の部分的な類似を検出する．

太田らの手法での問題点は文書間類似度を文間類似度の総和として定義していることに問題がある．そこで，本システムでは，文間類似度の分布を一種の画像として扱い，画像処理における代表的な直線検出アルゴリズムである Hough 変換とエッジ追跡を併用して類似部分を求める．これらのアルゴリズムを用いることで，局所的に文が入れ替えられている場合や，多少の文の追加にも対応することが可能となる．

本システムは各レポート間の類似部分とその類似度を求め,「どのレポートのどの部分がどの程度似ているか」を利用者に提示するだけでなく，この結果を元に参考・引用文献の検索支援を行い，検索された参考・引用文献と各レポートとの類似性も利用者に提示する．

以下では，まず本システムで検出することの出来る模倣レポートについて述べる．そして，本システムが用いている類似部分検出手法について説明する．最後に，現在のシステムにおける問題点についての議論を行う．

2　手法

本システムの処理の概要を図 1 に示す.

簡単に説明すると，システムは最初，入力されたレポート $d \in \mathcal{D}$ の全ての組 (d_i, d_j) に対して 2 文書評価を行い，類似部分とその類似度を求める．次にシステムは，複数の文書組で共通して類似している部分を抽出する．そして，その部分を元に生成した検索キーワードとレポート中に含まれる URL を用いて参考・引用文献の検索を行う．その後，入力レポート d と参考・引用文献 $r \in \mathcal{R}$ の組 (d, r) の全てに対して，前と同じ方法により 2 文書評価を行う．最後に，全文書に対する 2 文書評価の結果を用いて総合結果を生成する．

本システムが利用者に提示する情報は，個々のレポートに対する，被覆率ランキング，類似度ランキング，平均ランキング，参考・引用文献（以下，これらの情報をまとめて個別情報と呼ぶ）と全レポートに対する，総合被覆率ランキング，総合類似度ランキング，総合平均ランキングである．ここで,「被覆率」とは対象とするレポートの内容が，あるレポートにはどの程度含まれるかを表す指標である．そして,「類似度」とは，その類似の程度を表し,「平均」とはこれら 2 つの指標の調和平均を表す.

以下で各処理の詳細について述べる．

2.1　2 文書評価

本システムの核ともいえる部分である 2 文書評価の方法を述べる．図 1 からも明らかなように，本システムは 2 文書間評価を 2 回行うため，idf などの文書集合が全てそろわないと実行できない手法を用いると効率が悪くなってしまう．このような理由から，オンライン処理が可能な太田らの手法 [4] をベース

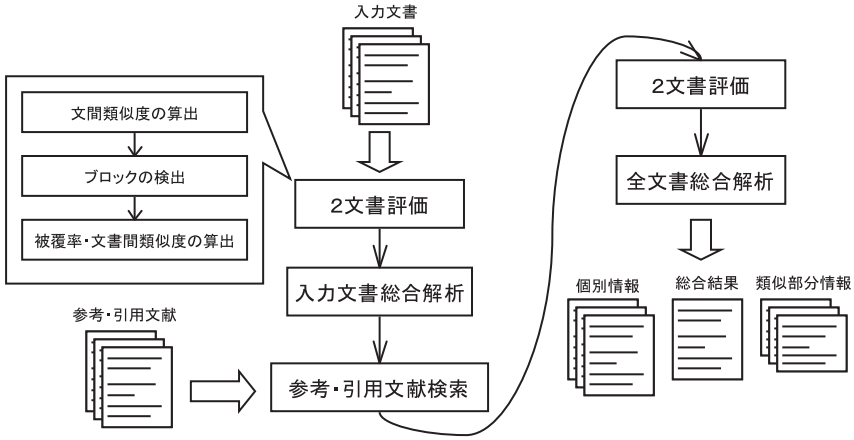

図 1: 処理の流れ

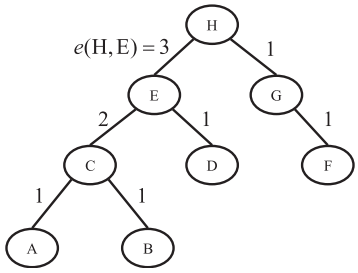

図 2: 枝の重みの例

とした．しかし，太田らの手法は類似部分を検出できないという問題点がある．そこで，類似部分を検出できるように改良を加えた．2 文書評価の処理の流れを示す．

step 1. 文間類似度の算出
2 文書評価の最初のステップは文間類似度の分布を求めることである．このステップで太田らの手法を用いる．

step 2. ブロック検出
次にその分布を用いて文書の「どの部分」が「どの程度」似ているかを求める（このようにして求められた類似部分を「ブロック」と呼ぶ）．

step 3. 被覆率・類似度算出
最後に，step 2 で求めたブロックから文書の被覆率と類似度（2.1.3 を参照）を求める．

以下では，2 文書評価の各ステップの詳細を述べる．

2.1.1 文間類似度の算出

太田らの手法には文の分割が発生した場合に各文間類似度が低くなってしまうという問題がある．これは，依存構造木の枝の重みが一様であることが原因である．そこで，本システムでは太田ら手法に改良を加え，文の分割が発生しても文間類似度が低下しすぎないように改良を加えた．

以下に具体的な改良点を示す．

- 枝の重みの設定
  文の分割に対して強くするために新たに枝の重みを設定した．語 w_i と w_j 間の枝の重み $e(w_i, w_j)$ を以下のように設定する．

$$e(w_i, w_j) = \begin{cases} \sum_k e(w_j, w_k), & w_k \text{ が存在するとき} \\ 1, & \text{それ以外のとき} \end{cases}$$

  ただし，ここで，w_i は依存構造木の根側の節点 (語) であり，w_k は w_j と直接繋がっている節点 (語) である．
  枝の重みの例を図 2 に示す．

- 正規化
  本システムではブロック（類似部分）以外の文間類似度は考慮されない．これにより，短すぎる文によるノイズを除去できるようになったので，文間類似度を正規化しても問題なくなった．これにより「文章の “流れ”」の類似度に焦点を絞ることができる．

以下に改良した文間類似度 $sim_{sen}(s_i, s_j)$ を説明する．今，文書 d を，

$$d =< s_i \mid i = 1, 2, ..., N_d >$$

とする．さらに，文 s は，

$$s =< ms_j \mid j = 1, 2, ..., N_s >$$

$$ms =< w_k \mid k = 1, 2, ..., N_{ms} >$$

とする．ここで，ms は最小構成文（文中で直接，もしくは間接的に係り受け関係となる語 w を集めた系列．詳細は文献 [4] を参照）であり，N_d, N_s, N_{ms} はそれぞれ，文書 d の文の数，文 s の最小構成文の数，最小構成文 ms の語の数である．このとき，文 s_i と s_j の類似度 $sim_{sen}(s_i, s_j)$ を以下のように定義する．

$$sim_{sen}(s_i, s_j) = \frac{sim_{sen}(s_i|s_j) + sim_{sen}(s_j|s_i)}{sim_{sen}(s_i|s_i) + sim_{sen}(s_j|s_j)}$$

$$sim_{sen}(s_i|s_j) = \sum_{ms_i \in s_i} \sum_{ms_j \in s_j} sim_{ms}(ms_i|ms_j)$$

$$sim_{ms}(ms_i|ms_j) = \sum_{k=2}^{|ms_i|} \frac{1}{\max\{W(w_{k-1}, w_k|ms_j)\}}$$

ここで，$W(w_{k-1}, w_k|ms_j)$ は最小構成文 ms_j において，語 w_{k-1} と w_k の間にある枝の重み $e(w_{k-1}, w_k)$ の集合である（ただし，w_{k-1} と w_k のいずれかが存在しない場合は $\{0\}$）．

2.1.2 ブロック検出

文間類似度の分布はある種の画像と考えることができる．すると，文書間の類似部分の発見という問題は，この画像での線分検出と考えることができる．線分検出を用いることで，DP などでは不可能な局所的な文の入れ替えや段落単位の入れ替えに対応が可能となる．そこで，本システムでは画像処理における幾何学図形の発見に用いられる代表的な方法である Hough 変換とエッジ追跡 [5] を併用し，類似部分（ブロック）の発見を行う．以下にブロック検出の方法を示す．

step 1. Hough 変換によるブロック候補の検索
ブロック検出の最初のステップは，Hough 変換を用いてブロック候補を検索することである．

step 2. エッジ追跡による仮ブロック検出
第 2 のステップは step 1 で求めた直線上の点で文間類似度が高い点から順にエッジ追跡を行う．追跡によって得られたブロックを仮ブロックとする．

ここで，ブロック b は，始点 $start(b) = (s_i, s_j)$ と終点 $end(b) = (s_{i'}, s_{j'})$ を持つ．始点はブロック b に含まれる文の組 (s_i, s_j) で i,j が共に最小となる点，終点は i,j が共に最大となる点である．さらに，ブロックの類似度 $sim_{block}(b)$ を以下のように定義する．

$$sim_{block}(b) = \sum_{(s_i,s_j)\in b} sim_{sen}(s_i, s_j)$$

step 3. ブロックの整理
最後のステップは step 2 で得た仮ブロックを整理して解となるブロック集合を得ることである．ここではまず，仮ブロックに変化がなくなるまで以下のルールに基づいてブロックの結合・削除を行う．

> **1.** 仮ブロック b_i, b_j において $end(b_i) = start(b_j)$ のとき，b_i と b_j を結合する．
> **2.** ブロック b_i, b_j のなす領域 $area(b_i)$, $area(b_j)$ が $area(b_i) \subset area(b_j)$ のとき，b_i を削除する．

そして，最後にブロックの角度 θ（始点と終点がなすベクトルの角度）が $\theta = 0$ or $\pi/2$ となるブロックを削除する．これによって得られたブロック集合を $B(d_i, d_j)$ とする．

画像処理では通常，線の検出には Hough 変換かエッジ追跡のどちらかを用いれば十分である．しかし，本手法では両方を用いている．これは，単純にエッジ検出を行った場合，線分を検出することは可能であるが，その処理の過程でエッジ点がずれる可能性がある．本課題では線分が 1 ピクセルでもずれるわけにはいかない．また，単純な Hough 変換では正確な位置での直線を検出することはできても，線分を正しく検出することは難しい．そこで，本システムでは Hough 変換とエッジ検出を併用することで正確な線分（類似部分）を検出するようにしている．

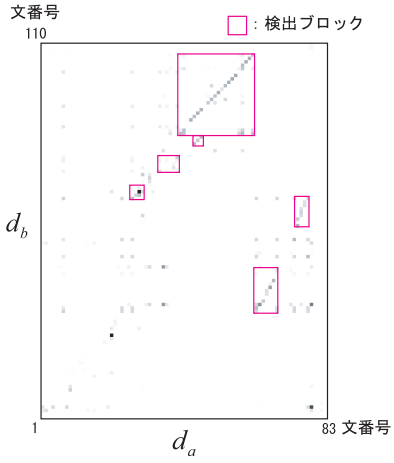

図 3: ブロック検出の例

ある文書 d_a と d_b に対してこの手法を適用し，取得したブロックの例を図 3 に示す．図 3 より，ブロック検出が正しく行われていることが確認できる．

2.1.3 評価

2 文書の評価は以下の 2 つの指標を用いる．

- 被覆率 $c(d_j|d_i)$
  d_i がどの程度 d_j をカバーするかを表す指標である．言い換えると「d_j において，d_i の内容が含まれている割合」である．これは以下の式で求められる．

$$c(d_j|d_i) = \frac{d_j\text{において } B(d_i,d_j) \text{ がカバーする文の数}}{|d_j|}$$

- 類似度 $sim_{doc}(d_j|d_i)$

$$sim_{doc}(d_j|d_i) = \sum_{b\in B(d_i,d_j)} \frac{|C(d_j|b)|}{|d_j|}\, sim_{block}(b)$$

$$C(d_j|b) = \{s \mid s \text{ は } d_j\text{において } b \text{ がカバーしている文}\}$$

2.2 入力文書総合解析

- 個別情報の生成 (ランキングの生成)
  入力文書 d_i に対して $c(d_j|d_i)$ のランキング，$sim_{doc}(d_j|d_i)$ のランキング，そして，被覆率と類似度の調和平均による平均ランキングを生成する（ただし，$d_j \in \mathcal{D}$ で $i \neq j$）．
- 類似部分情報の生成
  全てのブロックに対して以下のルールでクラスタリングを行い，類似部分情報を生成する．

> 2 つのブロック b_i,b_j が共通する文を含むとき b_i と b_j は同じ類似部分情報に属する．

表 1: 本システムによる結果 (被覆率, 類似度)

d_i \ d_j	1	2	3	4	5
1	1.0, 10.85	0.0, 0.0	0.0, 0.0	1.0, 7.74	1.0, 8.31
2	0.0, 0.0	1.0, 11.64	0.0, 0.0	0.0, 0.0	0.0, 0.0
3	0.0, 0.0	0.0, 0.0	1.0, 7.48	0.0, 0.0	0.0, 0.0
4	1.0, 7.74	0.0, 0.0	0.0, 0.0	1.0, 7.59	1.0, 6.21
5	1.0, 8.31	0.0, 0.0	0.0, 0.0	1.0, 6.21	1.0, 13.32

表 2: 文献 [4] の手法による結果

文書対	類似度	文書対	類似度
1-2	0.03	2-4	0.03
1-3	0.08	2-5	0.01
1-4	0.87	3-4	0.10
1-5	0.19	3-5	0.07
2-3	0.03	4-5	0.20

2.3 参考・引用文献の検索・取得

システムは入力文書総合解析が終わった後，参考・引用文献の検索・取得を行う．取得するべき Web サイトは

1. レポート中に記述されているサイト
2. 類似部分の文章が含まれているサイト

である．ここで，1 は直接ページを取得できるので問題ないが，2 は検索キーワードを自動的に生成する必要がある．本システムでは単純に，同じ類似部分情報を含む各入力文書で可能な限り長く，多くの文書で用いられている文字列をキーワードとする．[1]このとき基準となる指標は $length(s) \times sf(s)$ である．ここで，s は共通する文字列であり，$length(s)$ は文字列 s の長さ，$sf(s)$ は文字列 s の頻度である．

検索キーワードによっては検索結果が多数見つかる可能性がある．よって，検索結果をそのまま利用するのは現実的ではない．そこで，システムは検索を行った後，検索文書$\mathcal{R}$の上位 100 件までに対して次のような絞込み処理を行う．現在対象としている類似部分情報を S とし，これから検索された文書を $r \in R$ とする．このとき類似部分情報 S に属する文書 d と，検索された文書 r についての類似度 $sim_{doc}(d|r)$ を全ての組み合わせで求める．そして，この値の上位 n 件 (利用者が指定) を類似部分情報 S に対する参考・引用文献として採用する．

2.4 全文書総合解析

最後にシステムは全文書（入力文書と引用・参考文献）に対して総合解析を行う．このステップで追加・生成されるのは，個別情報の参考・引用文献と，総合被覆率ランキング，総合類似度ランキング，総合平均ランキングである．ここで，総合ランキングは参考・引用文献を含めた全ての文書組での，被覆率，類似度，調和平均のランキングである．

3 実験

本システムの手法の有効性を確認するために実験を行った．今回,「テレビゲームの中古販売」について書かれたレポートを 3 つ (文書 1~3) と，文書 1 を 2 名の被験者に「人のレポートを写すつもりで，読みながら写せ」と指示して作成した模倣レポート (文書 4,5) の計 5 文書を実験用文書集合として用意した．そして，この文書集合に対して本システムの手法と文献 [4] の手法による 2 文書評価を行った．実験の結果をそれぞれ表 1, 2 に示す．ここで，表 1 の各セルの値は $(c(d_j|d_i), sim_{doc}(d_j|d_i))$ である．

表 1,2 より，本システムの手法ではオリジナルとその模倣との関係にある，(文書 1, 文書 4),(文書 1, 文書 5) の組と，オリジナルの模倣同士の関係にある，(文書 4, 文書 5) の組を完全に捕らえることに成功している．しかし，文献 [4] の手法では (文書 1, 文書 5),(文書 4, 文書 5) の組で類似度が低くなってしまっている．これより，本システムの手法が模倣レポートの判別に有効であることが確認できる．

4 今後の課題

現在のシステムには以下のような課題がある．

- 計算コスト
  オンライン処理可能な手法を用いているが $O(n^2)$ の処理を何段にもわたって行うため，非常に計算コストが高い．
- 可視化
  現在システムは利用者にランキングの形で結果を提示している．この方法は直感的に判断が難しい

今後，これらの問題に取り組む必要がある．

参考文献

[1] 村田 哲也，黒岩 丈介，高橋 勇，白井 治彦，小高 知宏，小倉 和久，“学生レポートの n-gram による類似度評価の検討”，情報科学技術フォーラム (FIT) 2002 講演論文集，pp.101-102 (2002).

[2] 小川 貴博，岩堀 祐之，岩田 彰，“情報メディア教育における類似レポート判定システムの構築”，平成 13 年度電気関係学会東海支部連合大会講演論文集，604，p.304 (2001).

[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”，電子情報通信学会論文誌，Vol.J87-DII No.2，pp.661-672 (2004).

[4] 太田 貴久，増山 繁，“模倣レポート判定に用いる文書間類似度の考案”，言語処理学会第 10 回年次大会発表論文集，pp.729-732 (2004).

[5] 田村 秀行，“コンピュータ画像処理”，オーム社 (2002)

[6] “Google Web APIs - Home” `http://www.google.com/apis/`

[1]本来なら検索を自動的に行うようにするべきであるが，Google などの検索サービスを提供するサイトではプログラムによる自動検索を禁止している．そこで，Google Web APIs [6] を用いて自動的に検索を行う方法も考えられるが，Google Web APIs では日本語が扱えない等の問題があるため今回は検索キーワードの自動提示にとどまった．