3H2-OS3-8

文章の隣接グラフ化とグラフマッチングに基づく判例文の類似度計算

A graph matching procedure to find structural similarities from a legal case database

野坂 卓矢　　原口 誠
Takuya Nosaka　Makoto Haraguchi

北海道大学大学院 情報科学研究科 知識ベース研究室
Graduate School of Information Science and Technology Hokkaido University

This study aims at building a system to find similar legal cases, given a case description. Each case description is considered as a set of typed statements related each other by argumentation structure. So, in order to put different cases into correspondence, we consider a structure correspondence problem, as a kind of graph matching, under a constraint that the correspondence keeps statement types. We examine the ability of the approach for retrieving similar cases.

1. はじめに

1.1 本研究の背景

2010 年より, 裁判員制度が施行され, すでに死刑が請求されているような重大な事件に対する裁判員裁判も何例か行われている.

これにより, 一般市民も積極的に法律に触れる必要性が生じていると言えるだろう.

口頭弁論において, 原告や被告が自分達の正当性を主張する際に用いられる論拠の一つが, 判例である. これは, 過去に行われた自分達と類似した事例への裁判においての判決を例として持ち出すことで, その判決と同様の裁定を得ようとするものである.

本研究では, 一つの判例文を入力として与えたときに, 類似した判例文を検索するシステムを作り出すことを目的としている. ここでの類似した判例文の定義とは, 争点のオブジェクトである登場人物や場所は異なるが, それらのオブジェクト間で働いた現象, あるいはオブジェクトが起こした行動が多く共通しているもののことを指す. また,　裁判において判例を引用する際には, 現在争われている事例と類似しているだけではなく, その判例での採決を前例として現在の裁判にも適用してもらうために持ち出されるのが普通である. よって, 裁判所の判決も同じような判例が複数検索されるほうが望ましいと思われる.

本研究では, 上記の二点を考慮した文章間の類似度を定義することによって, 類似した判例文の検索の基準を制定しようと考えている.

1.2 判例文の特徴

判例とは, 一定の法律に関する解釈で, その法解釈が先例として, 後に他の事件へ適用の可能性のあるものである.

同様の事例に対して異なる判例がある場合, 優先順位としては, 上級審の判例が優先され, 同級審の判例同士では新しい判例が優先されるようになっている. 日本の場合では, 最上級の裁判所である最高裁判所の判例が最も優先される判例文となっている.

最高裁の判例文には, 最高裁の判断が書かれた「判断部」と呼べる段落がある. ここでは, 裁判所が下した判決の判断理由が述べられており, 裁判官が「事実部」に記述された内容をどう解釈していったのかの過程をたどることができる. また, ここでの記述はあくまでも裁判官の判断過程であり, 検証される事実の順序が実際の時系列通りに記述されていないという特徴がある.

連絡先: 原口　誠, 北海道大学大学院　情報科学研究科　知識ベース研究室, 北海道札幌市北区北 14 条西 9 丁目,011-706-7106,mh@ist.hokudai.ac.jp, http://www-kb.ist.hokudai.ac.jp

2. 文章の隣接グラフ化

2.1 類似度比較のための文章表現

本研究では, 文・動詞対-名詞ベクトルを作り, その自己相関行列を用いて文・動詞対間の関連性を表現する手法を提案する. この行列成分の値が大きいほど, 二文間が共通のオブジェクトに対する話題を扱っていると言える. この構造を隣接グラフで表現すると, 行列の成分の値が大きいほど, 対応するノード間の関連性が高いということになる. また, 共通の話題を持ち, かつ出現位置が近いノード間のエッジの重みが高くなり, 密集している構造になる.

また, 判例文の記述形式はある程度定められており, グラフの形状も似たような特徴が作られやすくなっている.

本研究は, 隣接グラフの形状を比較し, 類似した形状のノード同士をマッチングさせてから, ノードの文章における内容の類似度を比較させる. これによって, 話題の集合である密集地点同士の比較が可能となると思われる. 同様の手法を用いた判例検索システムもすでに提案されており [13], このようなアプローチはすでに提案されているものである.

この手法のメリットは,「時系列順に記述されていない判断部」を時系列を考慮せずにマッチングすることができたり,「オブジェクトは異なるが動詞は共通した事象群」を同一と判定できることである.

まず判例文の文章を形態素解析して, 各文ごとに出現した動詞と名詞を抽出する. 隣接グラフのノードにあたるのはここで抽出した一組の文番号と動詞の対である. 同じ名詞が出現している文のノードとノードの間には, 文番号が異なる場合にはエッジが張られる. エッジには重みがつけられており, ノード間の関連度の高さを表している.

まず, 行にはノードの基となる文番号と動詞を対にした組を, 列

にはエッジの基となる名詞を並べた文・動詞対−名詞ベクトル行列 F を作る.i 番目の文に動詞 v と名詞集合 N が出現していた場合,(i,v) 行 n 列の要素 $f_{(i,v),n}$ は,

$$k/|N|, k = \begin{cases} 1, & n \in N \\ 0, & \text{それ以外} \end{cases} \quad (1)$$

となる.
文中に含まれる名詞の総数が多いノードにペナルティをかける理由は, 名詞を多く含む長文は, 内容に関わらず他のノードとの間にエッジが必然的に張られやすくなってしまうため, 後の重要度判定に悪影響を及ぼすからである. その行列の自己相関行列 $F \times F^T$ を求めることにより, 各文と動詞の組同士のコサイン尺度を求め, その値を各文・動詞間の結合強度の元とする. さらに, ある行列要素の行の文番号が i, 列の文番号が j のとき,$e^{-|i-j|}$ を掛けることにより導出された値を, 各文・動詞対間の結合強度とする.
(i,v) 行 (j,w) 列の要素は,

$$e^{-|i-j|}\Sigma_N f_{(i,v),n} f_{(j,w),n} \quad (2)$$

となる.
文番号の差を取ることによって得られた互いの出現位置の距離を考慮するのは, 文章内での出現位置が近い文章は, 共通の話題を話している可能性が高いことを考慮するためである.
このようにして, 隣接グラフの隣接行列となる行列が求められる.
今回の研究では, この後の実験で用いる計算手法や計算時間の短縮のために使用するノード数を 50 個とする. 元の隣接グラフのノード数が 50 個に満たない場合は, エッジの張られていなく, 文番号や動詞も与えられていないダミーノードを追加することによって 50 個のノードを確保する.
また, ノード数が多くなった場合は, 隣接グラフを web ページのネットワーク構造と見なし,Google の Pagerank アルゴリズムを適用し, かつ判例文の判断部を表している手がかり文末表現を利用した一種の文書要約を行う [1]. その中で,Pagerank スコアの高くなった上位 50 個のノードを使用する.

2.2 予備実験

判例文を隣接グラフとして表現する方法の有効性について, スペクトルクラスタリングを用いた予備実験を行った.
まず, 二つの判例文文章についての隣接グラフ G と H を作り, 二つのグラフ G と H のノードを比較して, 動詞が一致したノード間には重さ 1 のエッジを, 互いの動詞が同義語の関係である場合には重さ 0.5 のエッジを張ることにより二つのグラフを一つに統合する.
グラフ G のノード (i, v_i) とグラフ H のノード (j, v'_j) 間に張られたエッジを表す行列 W を

$$W = [w_{i,j}] \begin{cases} 1, & v_i \text{と} v'_j \text{が完全一致} \\ 0.5, & v_i \text{と} v'_j \text{が同義語} \\ 0, & \text{それ以外} \end{cases} \quad (3)$$

とする.
G の隣接行列 A_G,H の隣接行列を A_H とすると, この G と H の統合グラフ U は

$$U = [u_{i,j}] \begin{pmatrix} A_G & W \\ W^T & A_H \end{pmatrix} \quad (4)$$

と表される.$d_{i,i} = \Sigma_k u_{i,k}$ となる対角行列を用いて, この隣接グラフのラプラス行列 $L = D - U$ を求める. この L に対し固有値分解を行い, 第一固有ベクトルの値が近いノード同士をマッチングさせるスペクトルクラスタリングを行った. このマッチング結果から, A_G の行と列の並び順を A_H に対応するように変える置換行列 P を生成するこの P を用いて本手法の類似度の指標とする

$$J(P) = \alpha \|P' A_G P'^T - A_H\|^2 - (1-\alpha) tr(P'W) \quad (5)$$

の値が求められる. 第一項は隣接行列のユークリッドノルムを求めることによってグラフの形状の違いを, 第二項は二つの文章間にどれだけ類似した動詞が出現しているかの指標となる. この値が小さいほど, 二つの判例文は相対的に類似した文章だということができる. また,α は第一項と第二項の優先度を調整する役割を持つ. この予備実験では $\alpha = 0.5$ とした.

2.3 予備実験結果

実験は, 文数が比較的小さな 4 つの判例文に対して行った. 集合には一つの被実験判例と, それと類似していると人の手で判断された正解判例があり, その他の判例は全て結果を比較するための比較判例とした.
そして, 被実験判例とその他の判例のそれぞれの類似度を本手法にて算出した.
予備実験は, 全て判例 A に対して行われる. 判例 A に類似しているとあらかじめ人手で判断した判例 B と, その他の判例 C,D とのそれぞれに対して J(P) を計算し, 判例 B の結果が判例 C と D の値より小さいかを確認した. その結果を表 1 に示す.

表 1: 予備実験結果

判例文 A との類似度	予備実験結果
判例文 B	0.0204
判例文 C	0.0836
判例文 D	0.2205

この予備実験の結果, あらかじめ人手で判例文 A と類似していると判断されている判例文 B の類似度の値が, 他の判例文 C や D に比べて小さい値を返すこととなった. これによって, 判例文のグラフ表現による類似度の計算が有効であることが示された.
この後の本実験では, 予備実験と同様の結果が得られるかを判断することによって, 本手法の有効性を示していく.

3. グラフマッチングによる類似度比較

3.1 提案手法

本手法は, 主に 3 つのステップからなる. なお, 判例文はすでに前処理が施され, 判断部を中心に重要なノード 50 個のみが集められた重み付きグラフの隣接行列が生成されているものとする. ノードには文番号と動詞の対と一対一で対応付けられている.
グラフ G と H の類似度を求める場合の本手法の手順は, 以下の通りである.

1. グラフ G と H の隣接行列 A_G と A_H を固有値分解し, 最大固有値に対応する固有ベクトル U_G と U_H の積 $U_H U_G^T$

を求め,50×50 の行列 Φ を生成する

2. G の各ノードに対応している動詞の集合 $V_G = v_1, v_2, \cdots, v_{50}$ を列に,H の各ノードに対応している動詞の集合 $V_H = v'_1, v'_2, \cdots, v'_{50}$ を行に配置した行列

$$W = [w_{i,j}] \begin{cases} 1, & v_i \text{と} v'_j \text{が完全一致} \\ 0.5, & v_i \text{と} v'_j \text{が同義語} \\ 0, & \text{それ以外} \end{cases} \quad (6)$$

を作成する

3. 行列 Φ と W を線形結合し, 新たな行列

$$\alpha\Phi + (1-\alpha)W, \qquad 0 \leq \alpha \leq 1 \quad (7)$$

を作成する.α はパラメータ値であるが, 本実験では $\alpha = 0.5$ としている. この行列にハンガリアン法を施し,G と H のノードマッチングを求める

上記の計算結果により, 動詞情報も考慮した置換行列 P' が求められる. この P' を用いて, 本手法の類似度の指標である

$$J(P') = \alpha \|P' A_G P'^T - A_H\|^2 - (1-\alpha) tr(P'W) \quad (8)$$

の最小値を計算する. 求められたこの値を,「判例文から作られた二つのグラフを類似した動詞を持つノード同士を多くマッチングさせた場合のグラフの類似度」の指標とする.

3.2 実験

実験は, 予備実験と同じ判例文 A,B,C,D に対して行い, 求められた J(P') の値を比較した. また, パラメータ値 α を変えることによる影響も調べた.

3.3 実験結果

この段落では, 本実験の結果について次の観点から検証を行う.

まず,J(P') の値が類似した判例文を判定できたか, 次に, パラメータ値 α の値を変えることによりどのような影響が出たかを検証する.

3.3.1 判例文集合間の類似度比較結果

表 2 は, 被実験判例とその他の 3 つの判例との類似度を求めるための置換行列を求め, それを基に被実験判例 A の隣接行列を置換した PAP^T と他の判例の隣接行列のユークリッドノルムの値を求めた結果である. 値の小さい判例文が, より被実験判例に類似していると考えることができる.

表 2: $\alpha = 0.5$ のときの判例文集合の類似度計算結果

判例文 A との類似度	本手法	第一項	第二項
判例文 B	-4.998796	0.001204	5
判例文 C	-2.917585	0.082415	3
判例文 D	-0.85227	0.14773	1

結果, 正解判例と比較判例との間に大きな値の開きが見られ, 予備実験と同様の結果を示した. これにより, 本手法により求めた類似度が文章間の類似性の指標として機能したことが確認できた.

3.3.2 パラメータ値 α を変化させた結果

先述したとおり, パラメータ値 α は隣接グラフの形状の類似度と文中の動詞の一致度の二つのうちのどちらを最終的な類似度により反映させるかを調整するためのものであり, 値が大きければグラフの形状の類似性を優先させたマッチングが行われ, 値が小さければ動詞が一致したノードが優先的にマッチングされる. この段落では, 実際にパラメータ値 α を極端な値である 0.9 と 0.1 に変化させた場合, 類似度にどのような変化が現れたかを確認している.

表 3 は隣接グラフの形状の類似度を優先させたものであり, 表 4 は動詞の一致度を優先させた結果である.

表 3: α=0.9 のときの判例文集合の類似度計算結果

判例文 A との類似度	本手法	第一項	第二項
判例文 B	-4.998796	0.001204	5
判例文 C	-1.919456	0.080544	2
判例文 D	-0.37774	0.12226	0.5

表 4: α=0.1 のときの判例文集合の類似度計算結果

判例文 A との類似度	本手法	第一項	第二項
判例文 B	-4.998796	0.001204	5
判例文 C	-2.917585	0.082415	3
判例文 D	-0.85227	0.14773	1

今回は二つの α の値を試したが,$\alpha = 0.1$ の結果が $\alpha = 0.5$ のときと同じになっていた. これは, 動詞が一致していた場合や同義語であった場合の配点が高く設定され, 結果として動詞情報に重点を置いたマッチングがされやすくなっていたためと推測さ思われる.

今後,α の値をもっと細かく変化させ, グラフ形状を重視したマッチングと動詞の一致を重視したマッチングの境界となる α の値を調べたり, 動詞が一致した場合の配点を調整する必要があるものと思われる.

4. まとめ

本論文では, 判例文文章間の類似性を捉える為, 判例文文章を隣接グラフ形式で表現し, グラフ間の類似度によって判例文文書間の類似度を評価する手法を提案した.

結果は, 提案した手法により求めた類似度は, 入力した文章に対して類似した文章とあまり似ていない文章との間に十分な差異が認められたので, 十分な効果があることを確認することができた.

今後は, 他のグラフの類似度を求める手法や判例の類似度を求める手法との結果の比較を行い, 本手法と他手法との得手不得手を調査する必要があると思われる.

特に, 今回用いた手法は互いのグラフのノード数を一致させる必要がある手法であり, 本研究では文章要約によるノード数の削減やダミーノードの追加によってその問題を解決した. しかし, そのようなノード数の調整をせずにグラフ間の類似度を求められる手法を導入することができれば, 文章要約に必要となっていた計算時間を削減させることができるだろう.

よって, 重み付きグラフの類似度を求める別手法の調査が今

後の課題である.

将来的には, この類似度を指標とした類似判例文の検索システムの構築に対してこの手法を応用したいと考えている.

しかし, この手法を実装する際の課題として, 判例文の類似度とノードマッチングにかかる計算時間が反比例していることが挙げられる.

初期状態にてマッチングしているノードが少ない場合, 徐々にマッチングの候補を増やしていく手法であるハンガリアン法は, 最初のマッチング数が少なければ最大でO (n^3) の計算時間を要する手法である. 逆にいえば, 計算に時間がかかっている判例文は類似していないとも取ることができる. よって, 実装の際には一定の計算時間を超えた場合にその判例文を類似していないと判定するなどの基準を設けることにより対処できる可能性があると思われる.

また, 全ての固有ベクトルを用いれば, 値の精度が落ちる代わりに計算時間が格段に速くなるので, そちらも対応策の一つとして考えられるだろう.

参考文献

[1] 大島 敦史, 原口 誠,"判例文の論理展開と意味的類似性に基く法律文要約手法の検討", 人工知能学会第 19 回全国大会講演論文集 (2005), 2G1-01.

[2] ウエストロー・ジャパン株式会社,"WESTLAW JAPAN"

[3] 法曹会,"最高裁判所判例解説 DVD"

[4] 有斐閣,"ジュリスト DVD", 創刊号〜1200 号

[5] SHINJI UMEYAMA, "An Eigendecomposition Approach to Weighted Graph Matching Problems", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 1988 sep, vol10, number5, pp695–703

[6] Miquel Ferrer and Francese Serratosa and Ernest Valveny, "Evaluation of Spectral-Based Methods for Median Graph Computation", Lecture Notes in Computer Science, 2007, vol4478, pp580–587

[7] Miquel Ferrer and Francese Serratosa and Albelto Sanfeliu, "Synthesis of Median Spectral Graph", Lecture Notes in Computer Science, 2005, vol3523, pp139–146

[8] Xiaoyi Jiang and Andreas Munger and Horst Bunke, "On Median Graphs: Properties,Algorithms, and Applications", "IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE", 2001 oct, vol23, number10, pp1144–1151

[9] メルビン A. ブルーア編 池田 敏雄校閲 林 孝雄訳, "ディジタル計算機の自動設計", 産業図書,1973

[10] David White and Richard C. Wilson,"Mixing spectral representations of graphs",18th International Conference on Pattern Recognition(ICPR'06)Volume 4 ,pp140–144,August 20-August 24,2006

[11] 大嶽 能久, 新田 克己, 前田 茂, 小野 昌之, 大崎 宏, 坂根 清和,"法的推論システム HELIC- ", 情報処理学会論文誌 vol35 No.6,June 1994

[12] Adam Schenker,Horst Bunke,Mark Last and Abraham Kandel,"Clustering of Web Documents Using Graph Representations", Studies in Computational Intelligence,vol52/2007,pp247–265,2007

[13] 原田 実, 鈴木 亮,"意味グラフのマッチングによる事故問い合わせ文からの判例検索システム JCare", 情報処理学会研究報告, 情報学基礎研究会報告,vol20/2001,pp15–22,March 2001

[14] Adam Schenker,Horst Bunke,Mark Last and Abraham Kandel, "Comparison of Algorithms for Web Document Clustering Using Graph Representations of Data", Lecture Note in Computer Science,vol3138/2004,pp190–197,2004

[15] 吉野 一,"法律人工知能", 創成社,2000