1C-03

隣接情報とグラフを用いた名称検索手法*

小川まな美† 佐藤正崇‡ 田山健一§

日本電信電話株式会社 アクセスサービスシステム研究所¶

1 背景

データベース (DB) における表記ゆれ (格納されているデータの表記が, 別の DB におけるデータの表記と異なる状態) は統合の際に, 表記ゆれを起こしたデータ同士が互いに同一の事柄を表現しているにも拘わらず, 異なるデータとして扱われる問題がある．表記ゆれは大きく 2 種類: i) データ名を省略した表記, ii) 使用者同士でのローカルルールに基づく呼び名 (通称) による表記, に分類される．従来の重複する部分文字列を含む表記を検索する手法 [1][2][3] は i) の省略表記のみが表記ゆれとして存在する場合に有効な手段である．一方で i) と ii) の表記ゆれが混在する状況に対処することは困難である．なぜならば通称表記は本来紐づけられるべき名称と著しくかけ離れているケースが多いためである．そのため, 2 種類の表記ゆれが混在する状況下で特に有効であるのは, 検索用の辞書を作成し同一の事柄を探し出す方法 [4] とされている．しかしながら, 突合する DB の個数が増加するとそれに伴い辞書を拡張する必要が発生するため, 辞書作成が完了するまでに時間が掛かるという欠点がある．

そこで 通信ビル間の経路を管理する DB では, 各データ (通信ビル) の物理的な隣接情報に加え, ビルを頂点としたパス情報が付与されていることに注目し, これらの情報を使用することで省略表記と通称表記が混在するデータを正確に紐づける手法を提案する．概要としては, 名称を紐づけたい 2 つの DB に対し一方の DB 内のデータ間には隣接情報が, 他方の DB 内のデータ間にはパス情報が付与されていると想定する．この下で一方の DB の隣接情報から得られるグラフ上で, 他方の DB のパスと同一の条件 (始点, 終点, 頂点数) を持つ経路を作成し, 最適な経路ともう一方の DB のパスの頂点同士を対応付ける．

接続情報の具体例としては, 表 1,2 のように各 DB に

データ ID	上位ビル	下位ビル
1	新宿ビル	代々木ビル
2	代々木ビル	千駄木ビル

表 1 隣接情報のある DB

「上位ビル」「下位ビル」という名前のカラムがあり、「上位ビル」に格納されたデータと「下位ビル」に格納されたデータはあるネットワーク上で隣接していることを表す．加えて表 2 のように 2 つの DB のうち 1 つには, 隣接情報に加えてパス (閉路) 情報が追加されている．

パス ID	構成ビル名
1	新宿ビル → 南新宿ビル → 外苑ビル
2	神宮ビル → 青山ビル → 渋谷ビル → 松濤ビル

表 2 パス情報のある DB

2 提案手法

2.1 付与情報によるグラフ構築

隣接情報が各データに付与されている DB(隣接 DB), パス情報が付与されている DB(パス DB) からビル名, ビル間の隣接情報、パス情報を各々得る．ここで隣接情報とは各ビル間の接続情報 (A ビルと B ビルはケーブルで接続されている等) を指し, パス情報とはパス DB 内のある 2 つのビルを始点, 終点としたパス (path) の情報 (X ビルから Z ビル, W ビルの順で Y ビルへ至る経路がある等) を指す．以下, 隣接 DB のビル名を原名, パス DB のビル名を略名と呼び, i 番目の原名を $b_i(i \in \{1,2,\dots n\})$, j 番目の略名を $d_j(j \in \{1,2,\dots m\})$ と書く．本稿では $n \geq m$ を想定する．

まず隣接 DB の隣接情報に基づき各原名を頂点, 隣接関係にある原名を辺で結ぶことにより無向グラフを作成する． 原名の集合を $V_b := \{b_i\}_{i=1}^{n}$, 辺の集合を $E_b := \{(b_i, b_h)\}$ とする． パス DB に付与されていたパス情報と, 始点, 終点及びパスの頂点数, を抽出する． 略名の集合を $V_d := \{d_j\}_{j=1}^{m}$, 有向辺の集合を $E_d := \{(d_j, d_k)\}$ とする． 隣接 DB より得られた無向グラフを $G_b := (g_b, V_b, E_b)$, パス DB 内のパスからなる有向グラフを $G_d := (f_d, V_d, E_d)$ とする．ただし $g_b : E_b \to P(V_b)$, E_bの元に V_bの部分集合を対応させる写像で, $P(V_b)$ は V_bの冪集合である． $f_d : E_d \to V_d \times V_d$, E_dの元に V_dの要素の対を対応させる写像．ここで, パス DB にパスはいくつあっても良いとするが, 次の 3 条件を満たすとする．i) 始点および終点に対応するビル名と同一の原名が存在する．ii) パスを構成する辺は全て隣接 DB より構成させるグラフに存在する．iii) 次数が 1 である頂点の接続している辺を除き, どの頂点および辺を 2 回以上通ることはない．

2.2 提案手法：パス情報を用いた頂点の対応付け

V_bと V_dで同一表記であるようなビル名の集合を S とする．各パスに対し以下 1)〜4) を適用する

1)G_dを構成するあるパスにおいて始点, 終点となっている頂点を $s, f \in V_d \cap S$ と表記する．また s, f を端点とする有向パスは

$((s, d_l), \dots, (d_k, f)) \in E_d^K$, K はパスを構成する辺の本数

*A method of searching strings using adjacent and graph information

†Manami Ogawa

‡Masataka Sato

§Kenichi Tayama

¶NIPPON TELEGRAPH AND TELEPHONE CORPORATION, Access service system laboratories

と表せるが, 本稿ではこのパスを

$$\Gamma := (s, d_l, \ldots, d_k, f) \in V_d^{K+1}$$

と表記する．ただし$\Gamma[k]$ をΓの第 k 番目の要素として $(\Gamma[k], \Gamma[k+1]) \in E_d$が成立する．
2)パスΓを構成する頂点の中で, 1) で定義した S に含まれる頂点の集合を

$$A := \{v_i | v_i \in S, v_i \in \Gamma\}$$

頂点集合 A の各要素のΓにおけるインデックスを．

$$I := \{x | \Gamma[x] \in A\}$$

と表記する．
3) グラフ G_bにおいて以下の 2 条件を満たすパスΓ'を作成する．
i)始点が s, 終点が f であり, 頂点数は $K+1$ である．
ii)パスΓ'を構成する頂点のうち, 頂点集合 A の各要素のインデックスは I と同一である．即ち

$$\Gamma'[x] = \Gamma[x] \in A, \Gamma'[x] : \text{パス}\Gamma'\text{の第 } x \text{ 要素}, x \in I$$

を満たす．
4)上記で得られたパスΓ'とΓを照らし合わせることで名称の組み合わせ

$$\{(b_i, d_j) | b_i = \Gamma'[x], d_j = \Gamma[x]\}$$

を得る．ただし 3) の 2 条件を満たすパスが複数存在する場合は全てを候補をして出力する．

3 数値実験

3.1 使用データ

通信事業者の業務において, 複数の通信ビル間ネットワーク (NW) を管理する通信パス構成管理 DB と, 光ファイバケーブル NW 構成管理 DB があり, 各管理ビル名に表記ゆれが存在するケースがある．加えて, 光ファイバケーブル DB はビル間の接続情報を保持しており, 伝送 DB の各ビルはエリアごとに閉路 (cycle) を形成している．これに注目し, 光ファイバケーブル DB を隣接 DB, 伝送 DB をパス DB として前述した条件を満たすパスを取り上げ, 72 個のビル名を実験に使用した．パス DB 内の各ビル名と同じビルを表す原名を, 提案手法を用いて隣接 DB 内のビル名から選択すると同時に, 精度比較として編集距離 (Levenshtein distance)[5] によるビル名同士の類似度から最も似ている名称を検索する実験を行った．

3.2 結果

既存手法と比べて提案手法では正解率が向上した. 各手法における正解数 (1 つの原名が正しく提示された略名数), 不正解数 (誤った原名が提示された略名数), 特定不可 (候補なし, または複数の略名数) を表 3 に載せる．

精度向上の理由としては編集距離による検索では原名と大部分が共通している略名が選択されたのに対し,

使用手法	正解数	不正解数	特定不可	正解率
編集距離	60	5	7	83.3%
提案手法	68	0	4	94.4%

表 3　各手法におけるビル名検索実験結果

提案手法では原名と全く共通する文字列がない通称であっても, グラフ上で頂点が一致することから正解の略名を選択可能であったからと考えられる．一方, 提案手法において特定不可となった名称は表 4 のように 2 通りの候補が出た．

原名の候補が少なくとも 2 通り存在してしまう原因は

略名	候補
x	a, d
y	b, c
z	c, b
w	d, a

表 4　提案手法において特定不可になった略名

, 表 4 の略名は全て 1 つの閉路の要素であり, この閉路を構成する頂点の多くが S に属していないこと (始点を除く 5 頂点のうち, 3 頂点目のみが S の要素) であったことだといえる．表 4 中の頂点が所属する閉路は$\Gamma :=$ (「U」,「x」,「y」,「V」,「z」,「w」,「U」) かつ, $S = \{U, V\}$ である．これより「U」を始点とし 4 頂点目が「V」であり頂点数 7 の閉路を作成するように提案手法を隣接 DB に適用すると (「U」,「a」,「b」,「V」,「c」,「d」,「U」) という閉路が出力される．ここで隣接 DB より得られるグラフは無向グラフであることから, 上記と逆順の閉路もまた「U」を始点とし 4 頂点目が「V」であり頂点数 7 の閉路という条件に合致する．よって表 4 のような結果が得られる．

3.3 今後の課題

100% の正解率で紐づけができる手法を作り出すことが今後の課題である．i) 今回の実験結果のように原名の候補が 2 通り出現してしまう場合の対処方法の構築, ii) パス DB のパスに関する 3 つ目の条件を, “次数が 1 である頂点の接続している辺を除き, 同じ辺を 2 回以上通らない” へ拡張すること, を目指す．

参考文献

[1] 中川 裕志 他, 「出現頻度と連接頻度に基づく専門用語抽出」, 自然言語処理, Vol. 10, No. 1, pp. 27–45, 2003.

[2] 田淵裕章, 他, 「N-gram に基づく用例対訳検索手法」, 信学技報, 人工知能と知識処理研究会, Vol. 108, No. 441, pp. 43–48, 2009.

[3] 小川まな美, 他, 「隣接情報を用いた文字列検索」, 電気情報通信学会総合大会講演論文集, 2019.

[4] https://www.fujitsu.com/jp/products/software/middleware/business-middleware/interstage/products/infoquality/

[5] D. Gusfield. “Algorithms on strings, trees and sequences: computer science and computational biology.” *Cambridge university press*, 1997.