4L-06

文脈を用いたHuman-powered結合処理方式の提案

中村 優太† 森嶋 厚行‡ 井ノ口 宗成†† 北原 格‡‡ 柚木 玲士†††

†筑波大学 情報学群 知識情報・図書館学類 ‡筑波大学 知的コミュニティ基盤研究センター

††静岡大学 情報学部 情報社会学科 ‡‡筑波大学 システム情報系

†††筑波大学 大学院 システム情報工学研究科

1 はじめに

人間を計算資源としてシステムに組み込むヒューマンコンピュテーションでは，計算機よりも人間が得意な作業を作業者に委託する事が一般的である．一例として，文字認識などのパターン認識を人間に委託するといった事が広く行われている．他の例としては，文脈を利用した同一性の判定がある．例えば，複数の人物の記述の実体の同一性判定において，“田中 太郎” と “Taro Tanaka” という情報だけだと，同姓同名の他人なのか，同一人物なのか判定が困難である．しかし，両者が執筆した論文のタイトルや所属が同じであると言う事がわかっていれば，人間は容易にこれらは同一人物の可能性が高いと判断可能である．

本稿では，マイクロタスクを利用したリレーショナル結合演算 (Human-powered 結合) において，文脈を用いる際の処理方式について議論する．上に挙げた例は，論文，所属，人物名などを含むデータの集合 R があるとすると，Human-powered (自己) 結合 $R \bowtie R$ とモデル化できる．Algorithm 1 に，Human-powered 結合の単純な処理方式を示す．これは，$ItemPairs = R \times R$ に含まれる各タプル毎に，順不同で結合判定のためのマイクロタスクを作成し，発行する．しかし，人が各マイクロタスクを行う際に，次のような問題がある．

- 結合判定のヒントとなる文脈が提示されない．
- 提示されたとしても，十分な文脈情報 (例えば，上記の例では，共著の論文など) がそろっていなければ，判定不能になる場合がある．

本稿では，文脈を用いた結合を行うための Human-powered 結合の処理方式である PuzzleJoin を提案する．PuzzleJoin では，結合判定の文脈の提示や，結合の判定に必要な文脈がこれまで得られたかどうかに応じて制御の切り替えを行う．PuzzleJoin の応用例の一つとして，自然災害時に撮影される多数の航空写真を貼り合わせる作業が考えられる．本稿では，実際の自然災害時に撮影された航空写真を利用して，PuzzleJoin を行った場合のシミュレーションを行った結果を示す．

Algorithm 1 単純な結合処理

```
Input: ItemPairs
Output: result
1: while ItemPairs ≠ ∅ do
2:     pair=ItemPairs.get
3:     generateTask(pair)
4:     if taskResult(pair) ≠ false then
5:         result.add(pair)
6:     end if
7: end while
```

関連研究． 論文 [1] は文脈を利用して，また，論文 [2] ではマイクロタスクを利用して実体同定を行う研究である．我々の知る限り，本研究は文脈を利用して人手で結合演算を行う処理に注目した初めての研究である．

2 PuzzleJoin

PuzzleJoin は，入力として，結合判定対象となるタプル集合 *ItemPairs* に加えて，次の 2 つの関数をとる．

enoughContext 関数． 結合判定を行うために十分な文脈がそろっているかを判定する関数である．引数として，タプル (r, s) とこれまでの結合の中間結果を取り，(r, s) の結合判定に必要な文脈が揃っていれば真を返し，そうでなければ偽を返す．

getContext 関数． 結合判定に必要な文脈の情報を取得する関数である．引数としてタプル (r, s) と結合の中間結果を取り，その結合判定に必要な文脈の情報を返す．

PuzzleJoin の基本的なアイデアは次の通りである．(1) enoughContext 関数を利用して結合判定に必要な文脈が揃っているタプルを見つけ，これらから順に判定のためのマイクロタスクを生成し，(2) そのタスクに，getContext 関数を用いて求めた文脈を表示する．

PuzzleJoin (Algorithm 2) では，*ItemPairs* が空になるまで次の処理を行う．まず，3 行目で，enoughContext 関数を用いて結合に必要な文脈があるか否かを判別する．4 行目で getContext 関数を用いて文脈を取得する．5 行目で，取得した文脈を用いてタスクを生成する．6, 7 行目で，判定が真であれば *result* に結果を追加する．

Proposal of a processing method of context-driven human-powered joins

†Yuta Nakamura, University of Tsukuba
‡Atsuyuki Morishima, University of Tsukuba
††Munenari Inoguchi, Shizuoka University
‡‡Itaru Kitahara, University of Tsukuba
†††Reiji Yunoki, University of Tsukuba

Algorithm 2 PuzzleJoin 処理

```
Input: ItemPairs, getContext(), enoughContext()
Output: result
 1: while ItemPairs ≠ ∅ do
 2:     pair = ItemPairs.get
 3:     if enoughContext(pair, result) then
 4:         context = getContext(pair, result)
 5:         generateTask(pair, context)
 6:         if taskResult(pair) ≠ false then
 7:             result.add(pair)
 8:         end if
 9:     else
10:         ItemPairs.add(pair)
11:     end if
12: end while
```

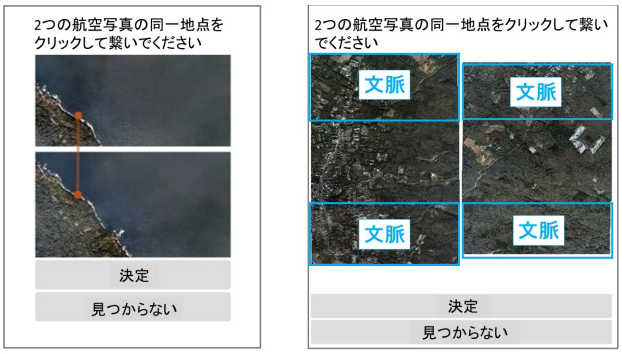

図 1: 文脈を利用しないタスク (左) と用いるタスク (右)

3 シミュレーション

複数の航空写真を結合し，1 枚の地図画像を生成する自己結合を行った．結合条件は，2 枚に重なりあう部分があるか否かである．

使用データ. 伊豆大島一部地域の航空写真データセット (平成 25 年台風 26 号伊豆大島災害後に撮影) [3] を使用した．図 2 のように上空から撮影されたものである．したがって，列方向に隣り合う画像同士の重なる領域は広く，行方向に隣り合う画像の重なる領域は狭い．航空写真は 308 枚あり，*ItemPairs* のサイズは 1470 である．

利用する文脈と結合判定タスク. 重なる領域に差があるため，本データセットでは行方向と列方向で組合せに必要な文脈が異なる．横の関係 (行方向) にある写真に対する結合判定においては，図 1 (右) のように各写真の列方向の上下 2 枚の結合情報を必要な文脈とした．すなわち，enoughContext 関数は，列方向に隣り合う画像同士の組合せである場合には無条件で真，行方向の組合せの場合には，上下二枚の結合情報が揃っている場合に真を返す関数とした．getContext 関数は，行方向の組合せが入力された場合に，その組合せそれぞれの上下二枚の結合情報を返す関数として定義した．列方向の組合せには文脈は不要のため null を返す．したがって，図 1(左) のような結合判定タスクとなる．

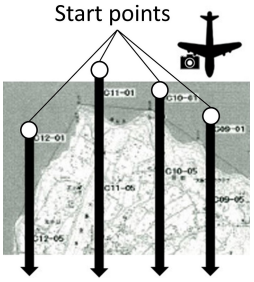

図 2: 航空写真撮影方法

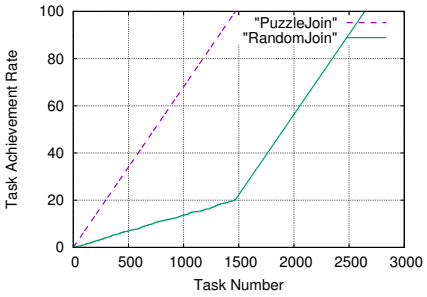

図 3: タスク処理シミュレーション結果

シミュレーション方法と結果. 作成した *ItemPairs*，getContext 関数，enoughContext 関数を利用して，次を比較した．(1) PuzzleJoin (2) getContext 関数を利用して文脈提示を行うが順序は制御しない単純な結合方式 (RandamJoin)．*ItemPairs* はキューとして実装し，文脈が不足しており判定できなかったタスクはキューの最後に追加した．その結果，PuzzleJoin は単純な結合処理と比較して約 44%少ないタスク数で処理できた (図 3).

4 おわりに

文脈を用いた Human-powered 結合処理方式である PuzzleJoin を提案した．今後は，必要な文脈の決定自体をクラウドソースする等の課題に取り組むことを予定している．

謝辞. 本研究の一部は科研費 (#25240012)，科学技術振興機構 SIP「レジリエントな防災・減災機能の強化」，および文部科学省「実社会ビックデータ利活用のためのデータ統合・解析技術の研究開発」の支援による．

参考文献

[1] Dong, Xin.; Alon, Halevy.; Jayant, Madhavan. Reference reconciliation in complex information spaces. Proceedings of the 2005 ACM SIGMOD international conference on Management of data. ACM, 2005.

[2] Zhang, Chen.; Jason, et, al. Reducing uncertainty of schema matching via crowdsourcing. Proceedings of the VLDB Endowment, 2013, 6(9) , p. 757-768.

[3] (データ提供) 東京都総務局総合防災部.