例示プログラミングを用いたデータ変換における曖昧さ解消のためのデータ中心型インタラクション

成田 穂 [*]　Nolwenn Maudet[†]　Yi Lu[‡]　五十嵐 健夫 [‡]

概要. 例示プログラミングは，反復的なデータ変換作業における手作業を減らす強力な手法である．しかし例示の数が少ないと曖昧さが残り，望ましくないデータ変換をしてしまうことがある．このような曖昧さは，生成されたプログラムをユーザが直接編集することで解決できるが，プログラミング未経験者にとっては難しい．そこで本研究では，例示によるデータ変換における曖昧さ解消手法として、データ中心型のインタラクションを提案する．これは，ユーザーがプログラムではなく出力結果を調べて修正することで，データ変換の曖昧さを解消するものである．これによりユーザーは生成されたプログラム自体を理解して書き換えるよりもはるかに簡単にデータ変換を行うことができる．本手法の重要な点は, 生成されたプログラムの汎用性や完全性を追求するのではなく，ユーザが変換したい特定のデータセットの処理に焦点を当てている点である．被験者実験の結果，本手法はプログラミング未経験者がより簡単かつ効率的にデータ処理するのに役立つことが示された．

1　はじめに

データ変換はデータ分析において重要な側面を占める [10][11][12][15]．しかしながら, データ変換は手作業による多くの試行錯誤を必要とし, 特にプログラミング経験のない人にとっては時間のかかる難しい作業である．彼らは処理後のデータがどう見えるべきかは分かっていても, それを実現するためのプログラムを書くスキルを持っていないことも多い．

例示プログラミング [8][13] は, プログラミング未経験者が簡単にデータの整形や変換ができるようにする強力な手法である．例えば, FlashFill [8] はユーザーが入出力例を与えると, それを実現する文字列変換ルールをドメイン固有言語 (DSL) として生成する．このような手法は便利であるが, 与えられた入出力例に対しては同じ結果を返すものの, 例にない別の入力に対して異なる出力を出すプログラムが複数生成されるという問題があり, これは生成プログラムの曖昧さとして知られている [13]．FlashProg [13] では曖昧さの解消のため, 自然言語で表現した生成プログラムの候補からユーザーが最も適切なものを選択する．しかし, データが複雑になると生成プログラムが冗長になったり, 類似するプログラムが大量に生成される傾向があり, FlashProg のようにプログラムを見て比較しながら曖昧さを解消する手法ではユーザーの負担が大きい．特に, ユーザーが手元にある特定のデータの処理だけに関心がある場合, 生成プログラムの汎用性や完全性を達成するための調整作業に長い時間をかけるのは非効率である．

そこで, 本研究ではプログラムの曖昧性を効率的に解消するために, 生成プログラムではなく変換後のデータを比較する「データ中心型」インタラクションを提案する．本手法では, 生成されたプログラム群を手元のデータにそれぞれ適用し, 各セルに対する出力結果の候補を作成する．出力候補が複数あるものを曖昧なセルと定義し, これがなくなるまでユーザーは作業を繰り返す．プログラムの出力結果を比較することで, ユーザーはプログラム自体を理解し修正するより, はるかに簡単かつ直感的にデータ変換を行うことができる．我々はセルの色付け，色付きのスクロールバー，チェックボックスという 3 つの機能を実装し, この「データ中心型」の対話的インターフェースを FlashAttention というインターフェースとして実現した．またプログラミング未経験者を対象に被験者実験を行って FlashFill との比較を行い, 本手法の有効性を検証した．

なお本論文は，ACM IUI 2021 で発表済みの内容 [14] をまとめたものである．

2　ユーザーインターフェース

図 1 に本手法の概要を示す．FlashAttention は現在テキストデータの変換に対応している． 元のデータは最初の行に表示され, ユーザーは 2 行目の “Output” に対応する出力例を与える．図の例では, 住所から番地（最初の数字列）を抽出したいとする．ユーザーはまず “9197 University St” から “9197” という入出力例を与える． すると FlashAttention

[*] University of Toronto
[†] University of Strasbourg
[‡] 東京大学

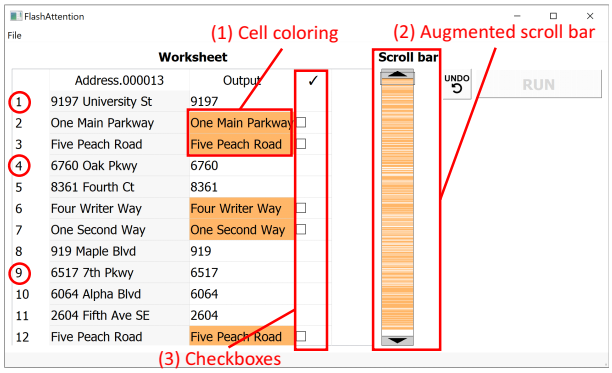

図 1. FlashAttention の概要. 本図はユーザーが一つ目の入出力例（“9197 Universiy St” から “9197”）を与え，チェックボックスを 2 回クリック（“6760 Oak Pkwy” から “6760”，“6517 7th Pkwy” から “6517”）した後の様子を示している．(1)-(3) は FlashAttention の各機能.

は対応するプログラムを生成し, 最も有力な出力候補を 2 行目に提示する．曖昧さ解消のために以下の 3 つの機能を使うことができる．1 つ目は曖昧なセルの色付けである．セルは複数の出力候補があるか, 候補が見つからない場合に曖昧と定義され, 橙色で強調される．2 つ目は曖昧なセルのある箇所が色付けされた拡張スクロールバーである．これによりユーザーは次に注目するべきセルを即座に見つけることができる．3 つ目は出力結果の横に置かれたチェックボックスである．これをクリックして一つ出力を確定させると, その出力を出すプログラム群だけが残されるため, その結果類似する出力を持つセルの変換結果を一度に全て確定できる．この他に, セルの出力結果を直接書き換えて入出力例を手動で追加することもできる．ユーザーはこれらの機能を用いて, 曖昧なセルがなくなるまで作業を継続する．例示プログラミングのアルゴリズムは FlashFill [8] と同じものを用いており, Generate, Intersect, Ranking の 3 つのフローから構成される．

3 被験者実験

FlashAttention と FlashFill を，各データ変換タスク完了にかかる所要時間およびエラー数に基づいて評価した．評価は、手法を被験者間因子（FlashAttention, FlashFill），データサイズを被験者内因子（Small, Medium, Large）として，2×3 の混合配置により行った． 被験者は，どちらか 1 つのインターフェースのみを使い，3 つのデータサイズに対して各 4 つ，合計 12 個のタスクに取り組んだ．各タスクは 1 つのカラムデータの変換を扱う．タスクセットのうち 7 つは Kaggle [2][4][5][6][7]，3 つは FlashFillTest [9] と呼ばれるベンチマーク，残りの

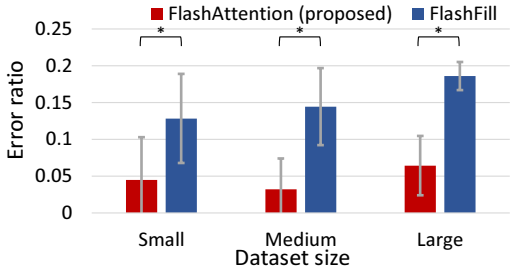

図 2. エラー率. Small(300 行)，Medium(3,000 行)，Large(30,000 行)．エラーバーは 95%信頼区間.

2 つは Web 上のデータをスクレイピングして取得した [1][3]．各タスクの難易度が可能な限り公平となるようにデータに一部変更を加えた結果, データサイズ Small は 300 行，Medium は 3,000 行，Large は 30,000 行のセルを持つデータとなった．実験の所要時間は参加者一人あたり約 60 分だった．参加者は計 26 名で, 全員が Excel など何らかのデータ処理タスクを毎週行っているが, プログラミング経験はない．

実験結果の検定は，2 元配置分散分析（ANOVA）により行った．タスク完了までにかかる時間は, FlashFill の 65.96 秒と比較して FlashAttention は平均 45.01 秒と有意に短かった（$F_{1,48}=25.64$, $p = 5.59 \times 10^{-20}$）．さらに FlashProg [13] と同様のやり方で，32 個の代表的なよくある間違い（例：欠損値の修正忘れ）を定義し，間違いの総数からエラー率を計算して評価した． 図 2 に DATASIZE ごとのエラー率を示す． FlashFill の 15.3%(SD=0.091) から, FlashAttention では 4.7% (SD=0.088) にエラー率が減少したことが分かった．この差は，エラー率に対する主効果（$F_{1,48}=29.99$, $p = 5.16 \times 10^{-22}$）に見られるように有意であった．DATASIZE とその交差には有意な主効果は見られなかった．これは FlashAttention が，データ変換作業中のユーザーのミスを防ぐのに役立つことを示している．

4 結論

本研究では，データ変換における曖昧さ解消のために, 予測された出力結果に注目するというデータ中心型のインタラクションを提案し, FlashAttention として実装した．被験者実験の結果，ユーザーはデータを効率的かつ正確に処理できることがわかった．

5 謝辞

本研究は JST CREST JPMJCR17A1 の支援を受 けたものである．

参考文献

[1] Auction Data, Gemini Trust Company. https://gemini.com/auction-data.

[2] Brazil Real Estate Listings. https://www.kaggle.com/devvret/brazil-real-estate-listings.

[3] NBA Advanced Stats. https://stats.nba.com/players/boxscores/.

[4] Suicide Rates Overview 1985 to 2016. https://www.kaggle.com/russellyates88/suicide-rates-overview-1985-to-2016.

[5] Titanic: Machine Learning from Disaster. https://www.kaggle.com/francksylla/titanic-machine-learning-from-disaster.

[6] Toronto Bikeshare Data. https://www.kaggle.com/jackywang529/toronto-bikeshare-data.

[7] Zomato Bangalore Restaurants. https://www.kaggle.com/himanshupoddar/zomato-bangalore-restaurants.

[8] S. Gulwani, W. R. Harris, and R. Singh. Spreadsheet Data Manipulation Using Examples. *Commun. ACM*, 55(8):97–105, Aug. 2012.

[9] S. Gulwani and E. Parisotto. PROSE Public Benchmark Suite, 2018. `https://github.com/microsoft/prose-benchmarks`.

[10] M. J. Kanter and K. Veeramachaneni. Deep feature synthesis: Towards automating data science endeavors. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, DSAA '15, Paris, France, 2015. IEEE.

[11] U. Khurana, D. Turaga, H. Samulowitz, and S. Parthasrathy. Cognito: Automated Feature Engineering for Supervised Learning. In *IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, ICDMW '16, pp. 1304–1307, Barcelona, Spain, 2016. IEEE Computer Society.

[12] T. Lau, S. A. Wolfman, P. Domingos, P. Domingos, and D. S. Weld. Programming by Demonstration Using Version Space Algebra. *Mach. Learn.*, 53(1-2):111–156, Oct. 2003.

[13] M. Mayer, G. Soares, M. Grechkin, V. Le, M. Marron, O. Polozov, R. Singh, B. Zorn, and S. Gulwani. User Interaction Models for Disambiguation in Programming by Example. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pp. 291–301, New York, NY, USA, 2015. ACM.

[14] M. Narita, N. Maudet, Y. Lu, and T. Igarashi. Data-Centric Disambiguation for Data Transformation with Programming-by-Example. In *26th International Conference on Intelligent User Interfaces*, IUI '21, p. 454–463, New York, NY, USA, 2021. Association for Computing Machinery.

[15] D. K. Wind. Concepts in predictive machine learning. Master's thesis, Technical University of Denmark, Copenhagen, Denmark, 2014.