

# 環境情報を考慮した GUI 操作のための文脈自由言語の学習

## Learning Method of Context-Free Language for GUI Operation with Background Information

秦野 亮\*<sup>1</sup> 東条 敏\*<sup>1</sup>  
Ryo Hatano Satoshi Tojo

\*<sup>1</sup>北陸先端科学技術大学院大学 情報科学研究科

School of Information Science, Japan Advanced Institute of Science and Technology

In this paper, we propose a method to learn a context-free grammar for GUI operation using background information. The method consists of three steps. First, we collect a example data as a raw-corpus from GUI users. The data is a sequence of pair of 1) an input-operation information which is represented by a symbols and 2) a background information which is represented by a labeled-ordered tree. Then we determine the terminal symbols of CFG. The symbols are defined by a set of pairs of an input-operation symbol and a symmetric difference of background information between current and previous states. Finally, we replace each input-operation informations of the raw-corpus to terminal symbols, and construct a context-free grammar by a grammar compression algorithm. If the ratio of the CFG rules of the result data is high, our method may be applicable to predictive input of input-operation without text.

### 1. はじめに

近年、携帯電話をはじめとする様々な計算機環境の利用において、効率的な入力を実現する様々な入力支援システムの存在は、欠かす事の出来ないものとなっている。例えば、仮名漢字変換、予測入力などのテキスト入力を支援するシステムである [小松 11][徳永 12]。

一方、GUI に対する「テキスト以外の」入力はどうか。例えばマウスクリック・ドラッグやタップ\*<sup>1</sup> などの入力である。GUI の場合、こうした入力の効率化には、例えばタスクに応じたインターフェースの改良 [Tidwell 11]、マクロ言語\*<sup>2</sup> [Yeh 09] の利用などの手段が取られる。ただし、うまく効率化がなされるかどうかは、人間が GUI に対してどのような入力行動を取るか、適切に分析して対応できるかによる。この入力行動は、ある特定の順番に沿って頻繁に行われる定型的な行動と、情報の探索など、試行錯誤を含むただ一度きりの目的を達成するために行われる非定型的な行動とに、大雑把に分類できよう。

ここで、GUI に対する入力行動のうち、定型的とみなせる行動について考えてみよう。我々は普段、画面に表示された内容を目で追いつながり、次の行動を決定している。このとき、画面を構成するコンポーネントには相当数の自然言語の文が含まれている。そして、その内容は次の行動を決定する際の指針となる場合が多い。こうしたことから、GUI に対する入力行動と、人間の言語処理的な知的側面との間には、何らかの関わりがあると仮定する事ができないだろうか。

ところで、人間の習慣的 (= 定型的) な行動を、形式言語の観点から分析を行った研究がある [三富 05][佐藤 08]。これらの研究では、人間の習慣的な行動規則を、文脈自由文法 (CFG) のクラスで取り扱っている。CFG とは、記号と生成規則の有限集合からなる形式言語モデルであり、生成規則は階層構造を

表現する事ができる。そのため、人間の習慣的な行動のうち、作法の様な階層構造を持つ一定の手順が存在するものをモデル化する事に向いているといえる。

では、計算機の GUI に対する定型的な行動は、CFG で十分にモデル化する事はできるだろうか。もし CFG で取り扱える割合が大きければ、自然言語処理の知見と併せる事で、より良い入力支援システムの開発が期待できる。例えば品詞  $n$ -gram モデルを用いた、GUI 操作に関する予測入力システムである。

ただし、CFG でモデル化をする際に、入力行動の情報を取り扱うだけでは不十分である。先ほど述べたように、我々は画面に表示された内容を基に次の行動を決定している。また、計算機上では、ある操作が、異なる場面で同じ結果をもたらすとは限らない。操作の結果、重大なエラーを引き起こすこともある。そのため、人間が「何をするか」という事だけをモデル化するだけでなく、操作対象に「何が起こるか」という情報を併せてモデル化する事もまた、重要であると考えられる。

本稿では、このようなモデル化の是非を検討するための CFG の学習手法について述べる。具体的には、人間の「計算機に対する操作」及び「操作によって GUI 環境上に生じた出来事」の情報を併せて、文脈自由言語 (CFL) のクラスで表現可能な定型的な行動を、CFG として学習する手法について述べる。

本稿の構成は以下の通りである。2 章では問題設定について説明し、提案手法の概要を述べる。3 章では CFL の基礎となる記号の学習手続きについて述べ、続く 4 章では CFG の学習について述べる。5 章では関連研究の紹介を行い、6 章で結論を示す。

### 2. 問題設定

GUI に対する定型的な行動を、文として生成可能な CFG を学習する問題について考える。学習の手続きは、おおまかに (1) 事例データの収集、(2) 終端記号の学習、(3) 構文規則の学習により構成される。詳細な定義、仮定等は後述し、本章では学習手続きの概要について述べる。

まず、学習の事例となるデータの収集を行う。事例データは、GUI への入力操作、各入力操作時点での GUI 環境を表す情報のペアの系列により構成される。次に、これらの事例データを

連絡先: 秦野 亮, 北陸先端科学技術大学院大学, 石川県能美市 旭台 1-1, r-hatano@jaist.ac.jp

\*<sup>1</sup> タッチ入力デバイスにおける、マウスクリックと同じ効果を持つ操作

\*<sup>2</sup> ソフトウェアによるキーボードやマウスなどの入力機器の入力自動化を行う事に特化したプログラミング言語・実行環境の一種

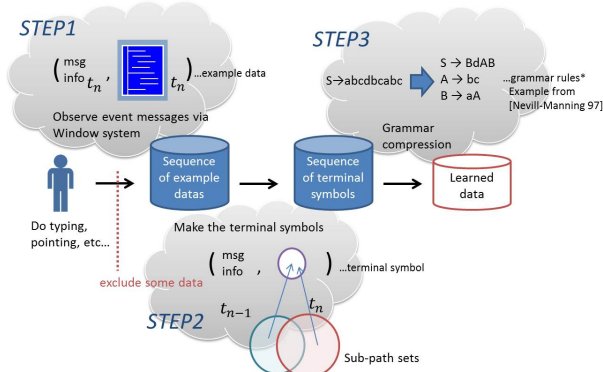


図 1: 学習手続きの概要

用いて CFG を構成する終端記号の学習を行う。終端記号は、入力操作を表す記号列、そして入力操作の前後で生じた GUI の変化を表す情報のペアとして定義される。そして、得られた終端記号で置換した入力操作の事例の系列を、CFG の構文規則を学習するアルゴリズムの入力とし、目的とする文法の出力を行う。以上の学習手続きを、図 1 に示す。

### 3. 終端記号の学習

#### 3.1 GUI 環境の定義

本稿では、GUI 環境を一般的なウィンドウ・システムと仮定し、そのウィンドウに関する情報をラベル付き順序木として定義する。GUI 環境は、ウィンドウやボタンなどの視覚的なコンポーネント、コンポーネント上に表示されるテキスト、その他コンポーネントが保持する属性値などにより構成される。そのため、GUI 環境はラベル付き順序木として形式的に表現する事ができる。

本稿では簡単のため、ラベル付き順序木の節点のラベルとして、コンポーネントのクラス情報<sup>\*3</sup>と画面上に表示されるテキスト情報を併せたものを用いる事とする。なお、以降の解説で断りなく「木」と表現する時は、このラベル付き順序木を表すものとし、文脈自由文法など他の木構造を表すものではないとする。図 2 下に、GUI の木構造での表現の例を示す。

#### 3.2 入力操作の定義

本稿では、GUI に対する入力を、ウィンドウ・システムに対するメッセージイベントのみと仮定し、操作の種類を識別する値、操作対象を識別するための値、操作に用いられた付加的な値のタプル (以下、操作タプルと記す) として定義する。GUI に対する入力操作は、計算機内ではメッセージイベントとして処理される。その際に、GUI のどのコンポーネントにどのような入力が行われたのかという情報が、メッセージデータに格納されて操作対象のコンポーネントを制御するプロセスに伝えられる。

本稿では簡単のため、このメッセージデータに含まれる様々な情報のうち、操作の種類を識別する値、操作対象を識別するための値、操作に用いられた付加的な値のみを用いる事とする。図 2 下に、入力操作のタプルでの表現の例を示す。

#### 3.3 事例データの定義

GUI に対する各入力操作について、その入力を表す操作タプルと、入力が行われた時点での GUI 環境を表す木構造によ

<sup>\*3</sup> ウィンドウ、ボタン、スクロールバーなどのコンポーネントの種類を識別する情報

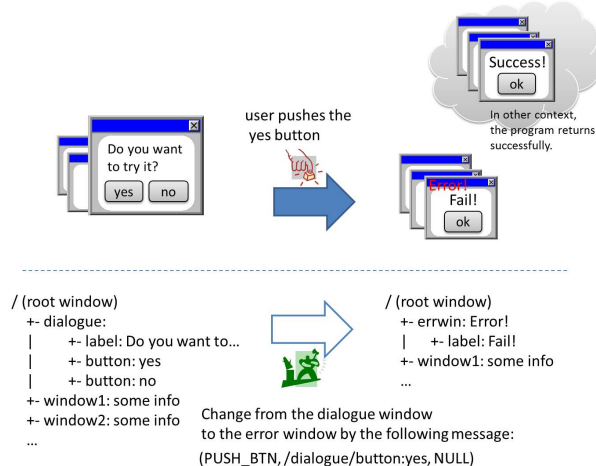


図 2: 実際の GUI と入力操作の形式的表現との対応関係

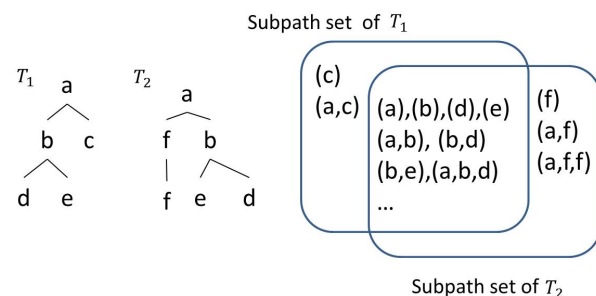


図 3: 部分パス集合の例

り構成されるペアを、操作事例と定義する。また、時系列順にソートされた全ての操作事例の系列を操作事例のコーパスと定義する。

本稿では、この操作事例のコーパスを用いて、CFG の終端記号及び構文規則の学習を行う。個々の終端記号は、操作事例の集合からの学習に基づいて定義される。この終端記号を学習するための事例を、終端記号の事例とする。終端記号の事例は、各入力操作の入力操作を表す操作タプルと、入力操作の前後で生じた GUI 環境の変化を表す木構造の差分により構成されるペアとして定義する。ただし簡単のため、この木構造の差分には、操作内容の識別にのみ特化した拡張部分パス集合を用いるものとする。

#### 3.4 部分パス集合の定義

部分パス集合とは、[Ichikawa 06] らにより提案された、木構造間の類似度の定義に用いられた考え方である。部分パスとは、木の根から葉までのパスとその一部の事である。また、部分パス集合とは木に含まれる全ての部分パスの集合の事である。図 3 に例を示す。この部分パス集合に基づく木構造間の類似度は、2 つの木の部分パス集合の共通部分について定義される。

#### 3.5 木構造の差分の定義

一方本稿では、この部分パス集合の定義を拡張して利用する事で、操作前後 2 つの木構造の差分を定義する。この差分には、入力操作によって増減した部分パスの情報が含まれている。この増減した部分パスは、2 つの木の部分パス集合の対称差集合に含まれる要素の事を指す。

しかし、部分パスはその定義からあるパスの一部であるという情報しか持たない。そのため、入力操作によって増減した部分パスが、元の木の異なるパスを起点としたものであったとしても、部分パス集合に変換する過程で同じものとして扱われてしまう恐れがある。即ち、異なる節点に対する操作を同一視してしまう可能性がある。

そこで本稿では、この木構造の差分を、木の根から変化の起点となる節点までのパスと、部分パスにより構成されるペアの集合 (拡張部分パス集合) として定義する。

### 3.6 終端記号の定義と学習

個々の終端記号は、終端記号の事例を用いて学習されると先に述べた。終端記号は、学習の結果、操作タプルと木構造の差分のペアとして定義される。異なる操作タプル・木構造の差分のペアは、異なる終端記号として扱われる。これは、ある操作タプルに複数種類の木構造の差分が対応づく可能性を考慮した事による。

GUI への入力操作は、異なる場面では異なる結果を伴う事がある。例えば、ある場面では既知の正常な動作をしても、別の場面ではエラーを起こす場合などである。本稿では、この様に見え同じに見える入力操作でも、異なる結果を伴う場合は異なるものとして扱う。そのため、終端記号を学習により定義するためには、ある操作タプルに関連する全ての終端記号の事例について、その差分の内容に応じたクラスに分ける必要がある。終端記号の木構造の差分とは、この各クラスに含まれるすべての差分の集合を指す。

### 3.7 木構造の差分のクラスタリング

ある操作タプルに関連する全ての終端記号の事例に含まれる木構造の差分を、内容の類似するもの同士で同じクラスとなるように、クラスタリングをする事を考える。本稿では、終端記号の識別のみを考え、クラスタリングの結果得られるクラスの数が適切となれば良いという立場を取る。そこで、任意の2つの木構造の差分を比較した時、小さい方の差分を基準として半分より多くの要素が共通する場合を同一のクラスとみなすという、単純な基準に基づくクラスタリングを定義する。なお、要素が空となる差分は、全て空要素のクラスとして扱う。

## 4. 構文規則の学習

CFG の構文規則を学習するために、オンラインの文法圧縮アルゴリズムである *SEQUITUR* [Nevill-Manning 97] を用いる。*SEQUITUR* アルゴリズムは、入力した区切りなしの文字列に対して、*Digram uniqueness* と *Rule utility* という2つの手続きを繰り返し適用し、入力した文字列を生成する事が可能な CFG を出力する。図4に各手続きの例を示す。生成される CFG は、2回以上同じ *digram* が CFG 中に含まれず、かつ全ての生成規則は2回以上用いられるという特徴を持つ。この文法圧縮アルゴリズムに入力する文字列として、本稿では事前に得られた終端記号で要素を置換したコーパスを用いる。

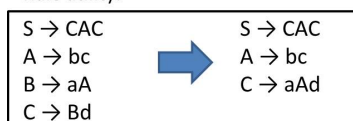
### 4.1 Digram uniqueness

*Digram uniqueness* では、生成規則を学習により追加する手続きが行われる。入力された文字列から文字を1つずつ読み込み、開始規則  $S$  に付け加えていく。この時、現在読み込んでいる文字とひとつ前の文字の組 (*digram* という) およびその出現位置を記録する。もし既に記録した *digram* と現在の位置の *digram* が一致する場合、非終端記号が割り当てられていなければ新たに非終端記号と *digram* を対応付けて生成規則とする。そして、現在の *digram* および以前に現れた *digram*

Digram uniqueness:



Rule utility:



Example from [Nevill-Manning 97]

図4: SEQUITUR の各手続きの例

を非終端記号で置き換える。また、既に非終端記号が割り当てられていれば、その非終端記号で現在の位置の *digram* を置き換える。

### 4.2 Rule utility

*Rule utility* では、1度しか用いられない生成規則の内容を展開して除去する手続きが行われる。これは、生成規則の数を減らし、その規則を含んでいた他の規則の内容をより長くする働きがある。

## 5. 関連研究

人間の習慣的な行動を形式言語的な側面から扱った研究として、映像分析に基づく習慣的な行動の認識 [三富 05] や、行動規則の文法を学習する [佐藤 08] という研究がある。[三富 05] らは、茶道の点前という、行動の構造が明らかな対象を認識するための CFG を与えており、それを動作認識タスクに利用している。また、[佐藤 08] らは、映像からの行動の文法を学習する際に低次の画像処理結果に含まれるノイズとなる記号を除去して、適切な記号列のみで文法規則を学習するという手法について述べている。

構文規則の学習に関する研究としては、左・右エントロピーに基づいた手法 [長尾 96] や、*Re-Pair* や *COMPRESSIVE* など他の文法圧縮アルゴリズムの研究が挙げられる。*COMPRESSIVE* は先に述べた [佐藤 08] らが用いていた学習アルゴリズムであり、*SEQUITUR* のオフラインヒューリスティック版という位置づけである。本稿では、GUI への入力操作が日々変化し、操作内容が増えていくという実用上の問題と対応がうまくように考慮したため、オンラインで取り扱える *SEQUITUR* を用いた。

## 6. おわりに

本稿では、人間の「計算機に対する操作」及び「操作によって生じた GUI 環境上の出来事」の情報を併せて、CFL のクラスで表現可能な定型的な行動を、CFG として学習する手法について述べた。学習した CFG は、今後 GUI に対する定型的な行動のモデル化として適当なクラスかどうか、実際に分析する必要がある。もしモデル化として適当なクラスであれば、より実用に即した木構造の操作手続きの検討を進めたり、より良い入力支援システムへの応用が期待できる。

また、本稿では終端記号の定義として入力操作、操作による木構造の差分を用いた。しかし、本来これらのペアは自然言語でいうところの形態素にあたと考えられ、単純に終端記号として定義すべきかどうかは検討の余地がある。もし、終端記号

として現状のペアではなく単語に相当する系列の定義が必要な場合、単語分割の問題を解決する必要がある。そのためには、アノテーションコーパスを用いた分類問題 [森信 11] を扱うか、統計的な深い洞察に基づく教師なしの単語分割 [持橋 09] の問題を扱うかというアプローチの仕方が必要になると考えられる。本稿では、こうした議論は避け、定型行動の構造を把握するための文法の学習手続きのみを扱ったが、この点は今後より一層の検討が必要な課題であると考えられる。

## 参考文献

- [Ichikawa 06] Ichikawa, H., Hakoda, K., Hashimoto, T., and Tokunaga, T.: Efficient sentence retrieval based on syntactic structure, in *Proceedings of the COLING/ACL on Main conference poster sessions*, pp. 399–406 Association for Computational Linguistics (2006)
- [Nevill-Manning 97] Nevill-Manning, C. G. and Witten, I. H.: Identifying hierarchical structure in sequences: a linear-time algorithm, *Journal of Artificial Intelligence Research*, Vol. 7, No. 1, pp. 67–82 (1997)
- [Tidwell 11] Tidwell, J.: デザイン・インターフェース 第2版 - パターンによる実践的インタラクションデザイン, オライリージャパン, 第2 (2011)
- [Yeh 09] Yeh, T., Chang, T., and Miller, R.: Sikuli: using GUI screenshots for search and automation, in *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pp. 183–192 ACM (2009)
- [佐藤 08] 佐藤洋一, 杉本晃宏 他: ノイズを考慮した人物行動の文法の教師なし学習, 情報処理学会論文誌. コンピュータビジョンとイメージメディア, Vol. 1, No. 2, pp. 86–99 (2008)
- [三富 05] 三富文和, 藤原冬樹, 山本正信, 佐藤泰介: 習慣的な行動の確率文脈自由文法に基づくベイズ識別 (画像認識, コンピュータビジョン), 電子情報通信学会論文誌. D-II, 情報・システム, II-パターン処理, Vol. 88, No. 4, pp. 716–726 (2005)
- [持橋 09] 持橋大地, 山田武士, 上田修功: ベイズ階層言語モデルによる教師なし形態素解析, 情報処理学会研究報告 (2009)
- [小松 11] 小松弘幸, 工藤拓, 田畑悠介, 向井淳, 花岡俊行, 湯川洋平: 予測入力システムの開発, コンピュータソフトウェア, Vol. 28, No. 4, pp. 17–22 (2011)
- [森信 11] 森信介, 中田陽介, Neubig, G., 河原達也: 点予測による形態素解析, 自然言語処理 = Journal of natural language processing, Vol. 18, No. 4, pp. 367–381 (2011)
- [長尾 96] 長尾 真, 佐藤 理史 (編): 岩波講座 ソフトウェア科学 [知識] 15 自然言語処理, 岩波書店 (1996)
- [徳永 12] 徳永 拓之: 日本語入力を支える技術 - 変わり続けるコンピュータと言葉の世界 (WEB+DB PRESS plus), 技術評論社 (2012)