

# 統合物語生成システムにおけるスクリプト型事象連鎖の獲得と利用

荒井 達也<sup>†</sup>

小野 淳平<sup>‡</sup>

小方 孝<sup>‡</sup>

岩手県立大学ソフトウェア情報学部<sup>†</sup>, 岩手県立大学大学院ソフトウェア情報学研究科<sup>‡</sup>  
 {g0311011, g236m001}@s.iwate-pu.ac.jp, t-ogata@iwate-pu.ac.jp

## 1. まえがき

統合物語生成システム (Integrated Narrative Generation System: INGS) (Ogata, 2016)において, ストーリー生成はストーリーコンテンツ知識ベース及びそれと対応するストーリー技法を利用して行われる. ストーリーコンテンツ知識ベースは各種事象連鎖に関わる具体的知識を格納する. その一つとして, Schank ら (1977, 1981, 1990)がスクリプトと呼んだタイプの事象連鎖知識も含まれる. これは, 人がある状況でどう行動するかを定義した事象連鎖知識である. 例えば, 「太郎がレストランに行く. 太郎がステーキを注文する. 太郎がステーキを食べる. 太郎が会計をする. 太郎が家に帰る.」のような意味的なまとまりの事象連鎖知識を言う. Schank らのシステムではスクリプトは人手により与えられていたが, 藤木・難波・奥村 (2002)は殺人事件という共通のトピックを持つテキスト集合から殺人事件に関するスクリプト知識を自動獲得する方法を提案した.

INGs のこれまでの研究は, 全体枠組み及び概念辞書の構築を主要目標として来たが, 現在, 上記のストーリーコンテンツ知識ベースの充実その他に研究の焦点をシフトさせている. ここでは, 人間行動におけるあるテーマに関する意味的にまとまりのある事象連鎖をスクリプトと見做し, その自動獲得と利用を目指す研究の現状を紹介する. これまで, 荒井・小野・小方 (2016)はスクリプトを半自動的に獲得・利用するためのツールを提案し, Ogata, Arai and Ono (2016)は, このツールを用いて作成したスクリプトを対象として 2 つの方法によるスクリプト知識獲得を試みた. 本稿では, これらを整理し問題点とその解決策を示すと共に, INGS を利用して作成したスクリプト知識を用いてストーリー生成を行う方法を示す.

## 2. 研究の背景

### 2.1. INGS

図 1 は INGS 全体の構成要素を示す. 物語生成はストーリー, 物語言説, 物語表現の 3 つの機構を通して行われる. INGS は, 事象連鎖知識等を格納するストーリーコンテンツ知識ベース, 事象の前状態と後状態を相互変換的に結び付ける規則を格納した状態-事象変換知識ベース, 動詞や名詞等の概念を格納する概念辞書その他の知識機構を持つ. 本研究が特に関連するのはこのうちストーリーコンテンツ知識ベースである. 各々のストーリーコンテンツ知識は, 二つ以上の動詞概念により構成される. 概念辞書を持つ動詞概念は, その概念の必須格を記述した格構造及び, 各々の格はその格が取り得る名詞に関する制約条件を持つ.

### 2.2. INGS におけるスクリプト知識の定義方法

INGs においては, スクリプト知識はストーリーコンテンツ知識ベース中のスクリプトコンテンツ知識ベースに

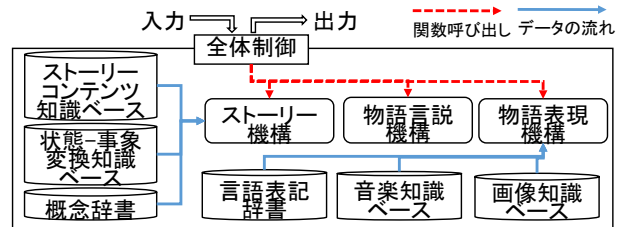


図 1 INGS の構成要素

格納される. スクリプト知識の実際の記述形式を図 2 に, 「男が食事する」という意味を持つスクリプト知識「食事する 1」の実例を図 3 に示す.

```
(scriptXXXX (<動詞概念> (<動詞概念 (1)><動詞概念 (2)>
... <動詞概念 (n)>))
((event <動詞概念> <格構造>))
(1 (event <動詞概念 (1)> <格構造 (1)>)
(2 (event <動詞概念 (2)> <格構造 (2)>)
...
(n (event <動詞概念 (n)> <格構造 (n)>))))
```

図 2 スクリプト知識の記述形式

```
(script0061 (食事する 1 (食べる 2 飲む 2 帰る 2))
((event 食事する 1 (agent (&v agel)) (location (&v loc1)))
((1 (event 食べる 2 (agent (&v agel)) (object (&v obj1)) (location (&v loc1))))
(2 (event 飲む 2 (agent (&v agel)) (location (&v loc1)) (object (&v obj2))))
(3 (event 帰る 2 (agent (&v agel)) (from (&v loc2))))))
```

図 3 スクリプト知識の例「食事する 1」

### 2.3. INGS におけるスクリプト技法

ストーリー技法における一タイプであるスクリプト技法は, ストーリー木の一部を拡張するスクリプト知識を用いる. 次のような方法でストーリー木を拡張する(図 4). 事象 E1, E2 から成るストーリー木 S に対し, 事象 E2 にスクリプト技法を適用することで, 事象 E3, E4, E5 から成る部分木を生成する. ストーリー木 S の事象 E2 を部分木で置換したストーリー木は S' となる. 図 5 はスクリプト技法の実際の記述を示す.

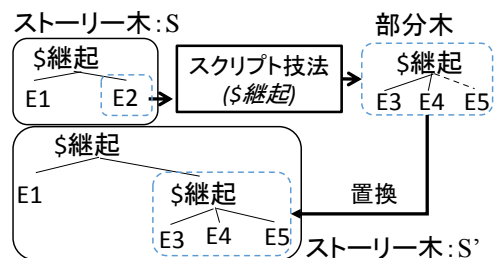


図 4 スクリプト技法の概念

```
(defun st-tech-scriptA (tar)
  (let ((relation '$継起) (eve-pos 'root) (type 'pat)
        (result nil))
    (setq result (st-tech-functempl tar relation
                                   eve-pos type))
    result))
```

図5 スクリプト技法の実際の記述

### 3. スクリプト半自動構築ツール

Common Lisp を用いてスクリプト半自動構築ツールを作成した。以下にツールを用いたスクリプト知識の構築手順を示す。

**Step.1 ユーザがスクリプト知識化対象の動詞概念を選択**

**Step.2 ユーザが対象の動詞概念を単文列で表す**

**Step.3 システムが単文列から動詞を抽出**

**Step.4 ユーザが抽出された動詞の格構造を決定**

**Step.5 システムが事象列の一貫性をチェック**

**Step.6 システムがスクリプト知識をスクリプト知識ベースに格納**

例として、「男が食事する」という意味を持つ動詞概念「食事する1」をスクリプト対象 (Step.1)とし、実際にスクリプト知識化する。

**Step.2 ユーザが対象の動詞概念を単文列で表す：**ユーザは、「男が食事する」という行動を3文以上で展開し、入力する(図6)。

「食事する1」を、三つ以上の文から成るスクリプトに展開してください。  
 なお、「食事する1」とは「男が食事する」という意味です。  
 入力をどうぞ。  
 >>男がご飯を食べる。男が酒を飲む。男が家へ帰る。

図6 ユーザによる「食事する1」をスクリプト展開

**Step.3 システムが単文列から動詞を抽出：**システムが、ユーザが入力した単文列「男がご飯を食べる。男が酒を飲む。男が家へ帰る。」を「。」で区別して形態素解析を行い、動詞「食べる、飲む、帰る」をそれぞれ抽出する。

**Step.4 ユーザが抽出された動詞の格構造を決定：**ユーザが、Step.3で抽出された動詞の意味を選択肢から選び入力を行う(図7)。この際、事象列に格構造が挿入される(図8)。

1:施行主が入出力で食べる  
 2:伯爵がステーキを食べる  
 番号を選択して下さい。  
 >>2  
 <<中略>>  
 (食べる2 飲む2 帰る2)

図7 ユーザによる格構造の決定

```
(event 食事する 1 (agent (&v age1)) (location (&v loc1)))
(event 食べる 2 (agent (&v age1)) (object (&v obj1) (location (&v loc1)))
(event 飲む 2 (agent (&v age1)) (location (&v loc1)) (object (&v obj2)))
(event 帰る 2 (agent (&v age1)) (from (&v loc2)))
```

図8 事象に挿入された格構造

**Step.5 システムが事象列の一貫性をチェック：**事象列中のそれぞれの事象に挿入された格構造に対し、以下の2つの方法を用いて一貫性をチェックする。どちらか一方に問題があった場合、Step.2に戻る。一つ目の方法は状態-事

象変換知識ベースを用いた事象の並びのチェックであり、二つ目の方法は格構造の制約条件のチェックである。

**Step.6 システムがスクリプト知識をスクリプト知識ベースに格納：**完成したスクリプト知識は、ストーリーコンテンツ知識ベース中のスクリプト知識ベースに格納される。2.2節の図3は、実際にスクリプト知識ベースに格納されたスクリプト知識「食事する1」である。

### 4. スクリプトの収集と形式化

276人の学生から回収した873個のスクリプトに対し、上記スクリプト半自動構築ツールを用いて860個のスクリプト知識を作成した。学生から回収したスクリプト(表1左)をINGRSにおけるスクリプト知識として作成する際、スクリプト半自動構築ツールへの入力として扱いやすいように、下の表2のような基準によりスクリプトを編集した(表1右)。スクリプト知識化が難しいと筆者らが判断したスクリプトが13個あった。図9は編集したスクリプト「料理する」からスクリプト半自動構築ツールを用いて実際に作成したスクリプト「料理する1」である。

表1 編集前後のスクリプト

回収したスクリプト	編集したスクリプト
タイトル：料理 スーパーで野菜を買った。家で野菜を洗った。野菜をフライパンで炒めた。野菜を皿に盛りつけた。出来た料理を食べた。	タイトル：料理する 男がスーパーで野菜を買う。男が家で野菜を洗う。男が野菜をフライパンで炒める。男が野菜を皿に盛る。男が料理を食べる。

表2 回収したスクリプトの問題と問題への対応

問題	問題への対応
タイトルが不明瞭	タイトルからスクリプト名を連想
動詞の過去形	動詞を現在形に変更
動詞の否定形	似た意味をもつ動詞に変更
修飾節等に含まれる動詞	事象一つにつき動詞一つとなるように単純化

```
(script1025 (料理する 1 (買う 2 洗う 2 炒める 1 盛る 2 食べる 2))
  ((event 料理する 1 (agent (&v age1)) (location (&v loc1)))
   ((1 (event 買う 2 (agent (&v age1)) (location (&v loc1)) (object (&v obj1))))
   (2 (event 洗う 2 (agent (&v age1)) (object (&v obj1))))
   (3 (event 炒める 1 (agent (&v age1)) (location (&v loc1)) (object (&v obj1))))
   (4 (event 盛る 2 (agent (&v age1)) (location (&v loc1)) (object (&v obj1))))
   (5 (event 食べる 2 (agent (&v age1)) (counter-agent (&v age2)) (location (&v loc1))))))
```

図9 作成したスクリプト知識「料理する1」

### 5. 収集したスクリプト知識の合成

作成したスクリプト知識の中で、格構造が挿入されていない事象を含むスクリプト知識を除いた、スクリプト知識ベース内の760個のスクリプト知識を使用して、以下の2つの方法により新たなスクリプト知識の作成を試みた。

#### 5.1. スクリプト知識の再帰的結合

スクリプト知識の最初の動詞概念から再帰的に組み合わせ処理を行い、スクリプトを拡張する。スクリプト内の動詞概念が別のスクリプト名称と等しい時、元のスクリプト知識は後者のスクリプトによって拡張される。なお本稿

では、一度使用されたスクリプト知識を再度使用しないものとした。図 10 に結合の手順を示し、実際に生成されたスクリプト知識の例を図 11 に示す。

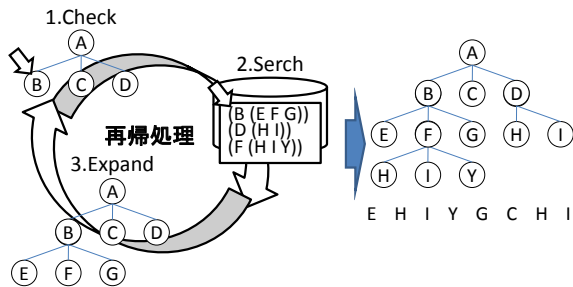


図 10 再帰的結合の手順

```
(scriptXXXX (起きる 4 (起きる 4 起きる 4 分かる 1
  食べる 2 出る 3 起きる 4 分かる 1 食べる 2 出る 3))
  ((event 起きる 4 (agent (&v age1)) (location (&v
    loc1)))
  ((1 (event 起きる 4 (agent (&v age1)) (location (&v
    loc1)))
  (2 (event 起きる 4 (agent (&v age1)) (location (&v
    loc1)))
  (3 (event 分かる 1 (agent (&v age1)) (location (&v
    loc1)) (to (&v loc2))))
  (4 (event 食べる 2 (agent (&v age1))
    (counter-agent (&v age2))
    (location (&v loc1))))
  (5 (event 出る 3 (agent (&v age1)) (from (&v
    loc4))))
  (6 (event 起きる 4 (agent (&v age1)) (location (&v
    loc1)))
  (7 (event 分かる 1 (agent (&v age1)) (location (&v
    loc1)) (to (&v loc2))))
  (8 (event 食べる 2 (agent (&v age1))
    (counter-agent (&v age2))
    (location (&v loc1))))
  (9 (event 出る 3 (agent (&v age1)) (from (&v
    loc4))))))
```

図 11 生成されたスクリプト知識の例

### 5.2. スクリプトネットワークからのスクリプト知識獲得

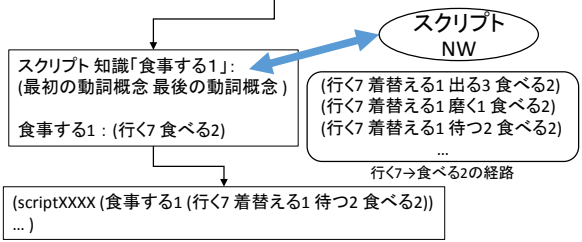
スクリプト知識のスクリプトをもとに重みを用いないスクリプトネットワークを構築した。このスクリプトネットワークの経路からスクリプト知識を作成する一手法を提案する。ここでは、スクリプト知識「食事する 1」を例に、具体的なスクリプト知識の作成方法を図 12 に示す。まず、既存のスクリプト知識ベースに格納されているスクリプト知識「食事する 1」は 6 つの事象「行く 7, 向かう 5, 買う 2, 食べる 2, 行く 7, 食べる 2」から成り、「食事する 1」は最初の事象「行く 7」から最後の事象「食べる 2」に向かって展開されている。スクリプトネットワークを対象に始点「行く 7」と終点「食べる 2」の経路を、網羅的に求める。求めた経路から経路長が 4 のものを選択し、格構造を挿入したものが新たなスクリプト知識「食事する 1」になる。図 13 は経路から作成した新たなスクリプト知識「食事する 1」の一つである。同様の手法を用いて、スクリプト知識ベース中の「乗る 3」から「降りる 1」で展開されるスクリプト知識「帰る 2」などの、他のスクリプト知識においても、様々な経路からなるスクリプト知識の獲得が望める。

### 5.3. ストーリー木への経路の挿入

スクリプトネットワークの経路を利用して、ストーリー木への事象列の挿入を行う。図 14 は、事象「食べる 2」

### スクリプト知識「食事する 1」

```
(script1008 (食事する 1 (行く 7 向かう 5 買う 2 食べる 2 行く 7 食べる 2))
  ...)
```



経路から作成したスクリプト知識「食事する 1」

図 12 スクリプト NW からのスクリプト知識作成の手順

```
(scriptXXXX (食事する 1 (行く 7 着替える 1 待つ 2
  食べる 2))
  ((event 食事する 1 (agent (&v age1)) (location (&v
    loc1)))
  ((1 (event 行く 7 (agent (&v age1)) (location (&v
    loc1)) (object (&v obj1))))
  (2 (event 着替える 1 (agent (&v age1)) (location (&v
    loc1)))
  (3 (event 待つ 2 (agent (&v age1)) (location (&v
    loc1)) (object (&v obj1))))
  (4 (event 食べる 2 (agent (&v age1)) (object (&v
    obj1)) (location (&v loc1))))))
```

図 13 経路から作成したスクリプト知識「食事する 1」

### ストーリー木

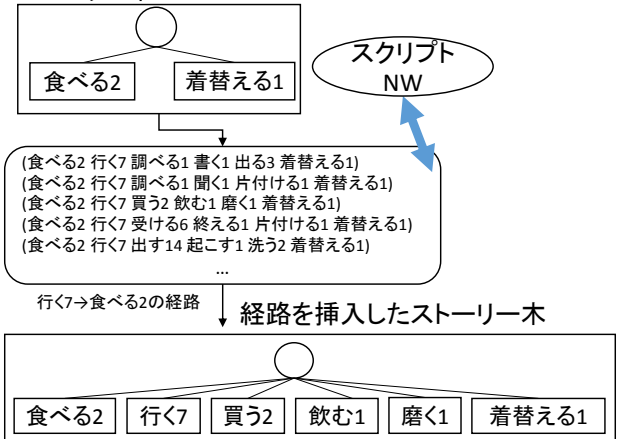


図 14 ストーリー木への事象列の挿入

と「着替える 1」の \$ 継起の関係から成るストーリー木に対し、スクリプトネットワーク中の「食べる 2」から「着替える 1」への経路を挿入してストーリー木を拡張した例である。図 15 は、ストーリー木 (図 15 上) と経路を挿入したストーリー木 (図 15 下) の実際の記述である。

### 5.4. 作成したスクリプトを用いたストーリー木の生成例

図 16 に以上の方法で用意したスクリプトの使用結果を示す。スクリプト使用前の太字部分が、スクリプト使用後の太字部分へと展開されている。なお、見易さのため事象の一部を省略している。実際の記述は次のようになる—  
 「(event 知る 1 (type action) (ID 8) (time (time7 time8)) (agent age%勇士#1) (counter-agent nil) (location loc%曲浦#1) (object obj%被害#1) (instrument nil) (from nil) (to nil) (adverb nil)

(possessive nil) (situation nil) (purpose nil) (experiencer nil) (source nil) (idiom nil) (information nil) (as nil)))」

## 6. 今後の課題

3節のスクリプト半自動構築ツールにおいて、現状の動詞概念辞書に登録されていない動詞概念がユーザによ

<pre>(\$継起 (event 食べる 2 (agent (&amp;v age1)) (counter-agent (&amp;v age2)) (location (&amp;v loc1))) (event 着替える 1 (agent (&amp;v age1)) (location (&amp;v loc1))))</pre>
<pre>(\$継起 (event 食べる 2 (agent (&amp;v age1)) (counter-agent (&amp;v age2)) (location (&amp;v loc1))) (event 行く 7 (agent (&amp;v age1)) (location (&amp;v loc2)) (object (&amp;v obj2))) (event 買う 2 (agent (&amp;v age1)) (location (&amp;v loc1)) (object (&amp;v obj1))) (event 飲む 1 (counter-agent (&amp;v age1)) (location (&amp;v loc1)) (instrument (&amp;v obj1))) (event 磨く 1 (agent (&amp;v age1)) (object (&amp;v obj1))) (event 着替える 1 (agent (&amp;v age1)) (location (&amp;v loc1))))</pre>

図 15 ストーリー木の実際の記述

使用前	<pre>(\$ロシア魔法音話 (\$問題 (\$予備部分 ...) (\$発端 ...)) (\$試行 (\$予備試練 (\$12 贈与者の第一機能 ...) (\$13 主人公の反応 (event 従う 1 (agent age%勇士#1) (location loc%登山鉄道#1) (object obj%命令#1)) (event 食べる 2 (agent age%勇士#1) (location loc%登山鉄道#1) (object obj%残飯#2))) ... (\$15 空間移動 (event 乗る 1 (agent age%勇士#1) (location loc%地方事務所#1) (object age%杖#1)) (event 浮かび上がる 2 (agent age%杖#1) (location loc%地方事務所#1) ...))))</pre>
使用后	<pre>(\$ロシア魔法音話 (\$問題 (\$予備部分 ...) (\$発端 ...)) (\$試行 (\$予備試練 (\$12 贈与者の第一機能 ...) (\$13 主人公の反応 (event 従う 1 (agent age%勇士#1) (location loc%登山鉄道#1) (object obj%命令#1)) (\$継起 (event 食べる 2 (agent age%勇士#1) (object obj%残飯#2)) (location loc%登山鉄道#1)) (event 行く 7 (agent age%勇士#1) (location loc%登山鉄道#1) (object obj%残飯#2)) (event 着替える 1 (agent age%勇士#1) (location loc%登山鉄道#1)) (event 出る 1 (location loc%登山鉄道#1) (instrument obj%星#1)) (event 見る 2 (agent age%勇士#1) (location loc%登山鉄道#1) (object obj%残飯#2)) (event 思う 1 (agent age%勇士#1) (location loc%登山鉄道#1) (object obj%残飯#2))) ) ... (\$15 空間移動 (\$継起 (event 確かめる 1 (agent age%勇士#1) (counter-agent age%老婆#1) (location loc%地方事 務所#1) (object obj%予約#1)) (event 開ける 1 (agent age%勇士#1) (location loc%地方事務所#1) (object obj%穴#1) (to age%杖 #1)) (event 開く 2 (agent age%勇士#1) (location loc%地方事務所#1) (object (age%杖#1))) (event 入る 2 (agent age%勇士#1) (instrument age%杖#1)) (event 座る 3 (agent age%勇士#1) (to loc%中 #1)) (event 閉める 1 (agent age%勇士#1) (location loc%地方事務所#1) (object age%杖#1)) (event 縮める 1 (agent age%勇士#1) (location loc%地方事務所#1) (object age%杖#1)) (event 浮かび上がる 2 (agent age%杖#1) (location loc%地方事務所#1) ...))))</pre>

図 16 スクリプト使用前と使用后

て入力された場合、格構造を事象に挿入することができないため、ツール側で類似する動詞概念を提示する必要がある。また、表記揺れにより動詞が検索できないことがあるため、言語表記辞書を利用し、表記揺れに対応する必要がある。

4節の INGS におけるスクリプトの収集と形式化では、上で述べた問題の影響により、格構造が挿入されていないスクリプト知識が 100 個存在した。これも同様に、半自動構築ツールの改善によりある程度の改善が見込める。また、今回作成したスクリプト知識はあくまで収集を目的としたため、スクリプト知識の持つ意味を評価していない。今後、スクリプト知識ベース中のスクリプト知識の意味の評価を行う必要がある。

5節では、760 個のスクリプト知識を利用して 2 つの方法から新しいスクリプト知識の獲得を試みた。5.1 節のスクリプト知識の再帰的結合では、スクリプト知識の結合を行い、5.2 節ではスクリプトネットワークの経路から新しいスクリプト知識を獲得した。双方の方法を用いてスクリプト知識の獲得を行う際、ある程度のスクリプト知識の長さを超えると、スクリプト知識の持つ意味が分からなくなる事態が発生した。今後、それぞれの獲得方法に応じた適切なスクリプト知識の長さを決めて獲得する必要がある。また、5.3 節のストーリー木への経路の挿入では、ストーリー木の事象の間にスクリプトネットワークの経路を挿入することでストーリー木を拡張した。今後、拡張したストーリー木を用いて、ストーリー生成を行う。例えば、あえて意味が捉えにくい程の長さをもつ事象列を挿入することで、ストーリーの印象を操作することが可能であると考えられる。

## 7. あとがき

以上、筆者らがこれまで行ったスクリプト知識の手動・半自動獲得、収集したスクリプト知識群から新たなスクリプト知識を自動獲得する方法を整理し、それぞれの問題点とその解決策を提示した。6 節で述べたスクリプト半自動構築ツールの改善と、獲得したスクリプト知識の意味の評価を行うことが今後の課題である。

## 参考文献

- 荒井達也・小野淳平・小方孝 (2016). 物語生成のための事象連鎖知識の半自動生成—統合物語生成システムにおける利用—. 『2016 年度人工知能学会全国大会(第 30 回)予稿集』, 3P1-1in2.
- 藤木稔明・難波英嗣・奥村学 (2002). テキスト集合からのスクリプト知識自動獲得. 『情報科学技術フォーラム—一般公演論文集』, pp.123-124.
- Ogata, T. (2016). “Computational and Cognitive Approaches to Narratology from the Perspective of Narrative Generation” in Ogata, T., and Akimoto, T. (Eds), *Computational and Cognitive Approaches to Narratology*, IGI Global, pp.1-74.
- Ogata, T., Arai, T. and Ono, J. (2016). “Using Synthetically Collected Scripts for Story Generation.” *Proceeding of The 26th International Conference on Computational Linguistics: System Demonstrations*, pp.253-257.
- Schank, R. C. (1990). *Tell Me a Story: A New Look at Real and Artificial Memory*, John Brockman Associates.
- Schank, R. C. and Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*, Lawrence Erlbaum.
- Schank, R. C. and Riesbeck, C. K. (1981). *Inside Computer Understanding: Five Programs Plus Miniatures*, Lawrence Erlbaum.