

スクリプトの 文章からの帰納的な学習

堀浩一 斎藤忠夫 猪瀬博

(東京大学工学部)

A method of inductive learning of scripts from natural language texts is given in this paper. A script is a stereotyped structure of contents of texts. An experimental system to learn scripts from texts called SEM (Script Extractor Mechanism) is described. The results of experiments of learning scripts from the texts of newspaper and of applying the learned scripts to classify texts are presented to testify the effectiveness of SEM.

1. まえがき

自然言語処理の研究は着実に成果をあげ、限られた分野の、限られた形式(タイトル、抄録など)の文章については、機械翻訳などの応用が実現されつつある。しかしながら、今後、第5世代計算機の研究成果などもふまえ、自然言語処理システムを、広く知識情報処理システムの一環としてとらえる時には、まだ残されている問題は多い。特に、文章の意味の理解の問題は、いざいざ通ることで済む問題である。機械翻訳においては、表層構造の取扱いに重点を置いて、限られた応用では充分実用に供する事が期待できるが、抄録の作成、知識ベースやデータベースとのインタフェイスなどの応用においては、自然言語処理システムそのものが充分な知識をもっていて、文章の意味を理解できるようにしなければならない。意味の取扱いが重要であり、自然言語理解システムに知識を付与することにより、その取扱いが可能となりうるのであるということとは従来から重ねて指摘されてきた。それにもかかわらず、期待されるほど研究が進展しないのは、自然言語理解システムに与える知識作成の困難さに起因していると筆者らは考える。筆者ら自身、以前に、論文抄録の文章が内容の定型的な構造をもつことに着目し、その構造の知識を利用して、抄録の文章からキーワードの抽出を行うシステムを作成し、実験した¹⁾。このシステムは、かなり良好な実験結果をおさめたが、知識の作成、修正、保守の困難のために、大規模な実用システムの作成に至っていない。

この問題を解決するためには、自然言語理解システムが必要とする知識を文章そのものから学習システムによって学習することを考える必要がある。自然言語理解システムに必要なとする知識には、辞書、文法規則、対象分野の専門知識、スクリプトやシナリオなどと呼ばれる文脈の構造に関する知識などが含まれる。これらすべての知識について、作成や保守の手法が考えられておかねばならない。本論文では特に、文章の内容の定型的知識であるところのスクリプトを、文章から帰納的に学習する手法について論ずる。スクリプトが自然言語理解システムにとって有効であることはよく知られており、また辞書や文法規則に比べれば変化に乏しいので実用的価値のある学習結果を得ることが可能であると期待されるからである。2章で、スクリプトの表現法と利用法と帰納的学習の方法について論じ、作成した実験システムSEM (Script Extractor Mechanism) について説明する。3章で新聞記事の文章を対象とした実験について述べ、スクリプトの帰納的学習の有効性と限界を示す。

帰納的学習は、実例を与えることにより、実例が共有している性質あるいは原理を学習させようとするものであり、従来、帰納的推論のアルゴリズムの研究²⁾、類推の理論の研究³⁾、Meta-DENDRAL⁴⁾やBacon⁵⁾などの特定の分野の学習システムの作成、文法の推論の研究などが行われてきている。従来の研究については文献6)に詳しくのべられているが、帰納的学習の一般論に重点を置いたシステムでは実用性に欠け、実用性に重点を置いたシステムでは、問題特有の知識が学習プログラムにうめこまれてしまつて汎用性に欠けるという傾向をもっている。本論文で提案するSEMは、実用性をめざしつつ、問題特有の知識は分離して記述することにより汎用性、拡張性に留意している。また、学習の目標であるスクリプトは、正解が1つに定まり、このわけではなく、スクリプトの使用目的に応じて、別の形のスクリプトが必要なことも考えられる。そこで、帰納的学習を行うための規則もプログラムとは分離して記述することにより柔軟性をもつように考慮している。

2. スクリプトの帰納的学習と利用

2.1 スクリプトの表現法と利用法

スクリプトは、当初、Schankにより提案された、レストランでの食事などの人間の日常の行為の定型の記述である⁷⁾。同じくSchankにより、スクリプトを自然言語の文章に適用して文章を理解するSAM (Script Applier Mechanism) と称するシステムが作成された⁸⁾。Schank自身はその後、スクリプトはダイナミックに構成されるべきであるとし、その構成要素としてMOPs (Memory Organizing Packets) という考えを提案している⁹⁾。本論文では、スクリプトという用語は文章の内容の定型の記述という意味で用い、必ずしもSchankの研究にはとらわれぬことにする。最終的には、Schankと同様に、MOPsとスクリプトの階層を考え、複雑な理解システムを構成することが目標である。しかし、MOPsの学習やスクリプトのダイナミックな構成の研究は未知の分野であり、まずは理解システムの行う仕事としては簡単なものと設定して、それに必要な比較的簡単なスクリプトの学習の研究から着手すべきであろう。そこで、本論文では、スクリプトを用いて文章の分類を行うことを、自然言語理解システムの行う仕事として設定する。分類の次に、キーワードの抽出システム、その次に抄録の作成システムなどを想定している。

システムの全体像は図1のようになる。たとえば、新聞記事の文章を、日米経済摩擦に関する記事とそうでない記事に分類したいとしよう。すると、ユーザは、SEM (Script Extractor Mechanism) に、日米経済摩擦の記事の実例をいくつか与える。SEMが帰納的学習を行って、日米経済摩擦とはこういうものだというスクリプトを出力する。SAM (Script Applier Mechanism) が、そのスクリプトを新しい文章に適用して、日米経済摩擦にあてはまるかどうかを判断する。そこで、SEMの出力するスクリプトは直接人間が見ても意味がわかるものでなければならぬとする。

スクリプトの表現方法としては、筆者らの提案したH-net (Hierarchical network) と称する知識表現法を用いる。H-netは、概念の外延と内包を論理的に表現する

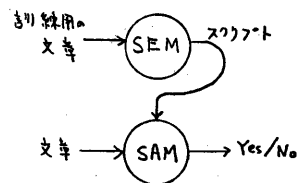


図1: システムの全体像

る知識表現法である。H-net を用いると、スクリプトからシソーラスまで一貫して論理的表現ができる。表現の意味と明示することができる。たとえば、日米経済摩擦のスクリプトが、「アメリカが日本と非難した。アメリカは $\times X$ を言った。 $\times X$ は経済問題である。日本は $\times Y$ の対応をとった。」というものであり、たとえると、このスクリプトは図2のよう表現できる。さらに、経済問題とはどういうものか、対応とはどういう動作かというシソーラスや、格構造のような動詞に関する知識が存在して、やはり H-net で表現されるが、ここでは予じめ与えられているものとする。図2で $\times text$ は、文章にバインドされる変数である。(非難 $\times text$ アメリカ 日本) は、 $\times text$ 中に、アメリカが日本と非難したという記述があることを示している。本来、スクリプトは時間的前後関係を含んでいるが、新聞記事の文章では、どこどこと時間関係と文の前後関係は一致せず、いろいろな順序で出現するので、図2のスクリプトには、時間的前後関係は含んでいない。時間の前後関係を扱うには、各述語に、調べた残りの文章をバインドする変数を付加して次々に理解できるようにすればよい(たとえば、(非難 $\times text$ $\times rest1$ アメリカ 日本) (言う $\times rest1$ $\times rest2$ アメリカ $\times X$) というようにする)。

日米経済摩擦

```
intension
(lambda (xtext)
  (非難 xtext アメリカ 日本)
  (言う xstart アメリカ xX)
  (経済問題 xX)
  (対応 xtext 日本 xY))
```

図2 スクリプトの H-net による表現の例

2.2 帰納的学習の方法

スクリプトは次のようなステップをへて学習される。

Step 1. 文章を愚鈍な理解システムで理解してイベント・ストラクチャと呼ぶ構造を得る。

Step 2. イベント・ストラクチャ相互のマッチングをとる。(3つめ以降の文章については、文章のイベント・ストラクチャとスクリプトのマッチングをとる。)

Step 3. マッチングのとれた部分ととれなかった部分をルール・セレクトに渡して一般化のためのルールを選択する。

Step 4. 一般化のルールを適用して、構造を一般化する。

Step 5. スクリプト再構成のためのルールを適用してスクリプトを再構成する。

Step 1 から Step 5 をくり返す。

各ステップの詳細を例を用いて説明する。

今、「Aという人がBという人を殺し、Aという人は自殺した」という新聞記事の文章のスクリプトを学習させたいとする。ユーザは図3に示すような実例を入力する。これを愚鈍な理解システム(parser といってよい)が解析して、図4のようないイベント・ストラクチャを出力する。ここでは、イベント・ストラクチャは、述語、主語、目的語からなる構造としておく。(理解システムが賢くなるにつれてイベント・ストラクチャも精密にするとは将来の課題とする。) また、照応の取扱いは、愚鈍な理解システムはできないものとする。次に、このイベント・ストラクチャどうしのマッチングを述語名についてとる。(3つめ以降の文章とスクリプトとのマッチングの場合は文章のイベント・ストラクチャがスクリプトに含まれるかどうかと調べる。) この例の場合、(殺す 調理師 幸枝さん)と(殺す 自分 二人)

最終的なスクリプトが得られる。この再構成のステップは現在はまだ実装されていない。

以上の説明から明らかなように、SEMでは、スクリプトにあてはまる実例だけを訓練例として与えて学習を行う。あてはまるない例についても教える方法も考えられるが、Winstonのいうニアミスを与える必要があり³⁾、実際の文章からニアミスを選んでは与えるのは面倒である。また、人工的なニアミス例をつくると、それは正解を知っているからこそのできることであり、本論文の趣旨に反する。あてはまる例だけからの学習を図式的に示すと図6のようになる。xを実例、実線の円を正解、破線とSEMの出力とすると、xを囲むように、破線の円を少しずつ広げて、正解に近づけようとするのがSEMの学習のやり方であるといえる。

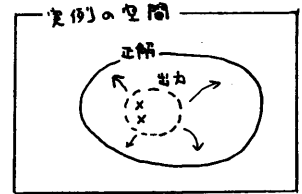


図6 SEMの学習の図式的説明

2.3 実験システム SEM (Script Extractor Mechanism)

前節で述べた学習を実現するための、SEM 1.5 と称する実験システムを、東京大学大型計算機センタ上に、Franz Lisp と等価の作成した HNRL (H-net Representation Language)¹⁰⁾を用いて実装した。

SEM 1.5 の構成を図7に示す。実際に実装されているのは、図中で、実線で囲んだ部分である。dull-understander は現在実装されておらず、人間がイベント・ストラクチャを作成している。また、reconstruction のステップは素通りしている。四角で囲んだ部分は Lisp により、丸く囲んだ部分は H-net により記述されている。

matcher は出現した述語名にマージをつけることにより、同じ述語名の述語を見つけ、マッチした述語とその周辺の述語からなる構造を rule-selector に渡す。

rule-selector は前節に述べたとおり、述語を一般化するためのルールを generalization-rules の中から選り出す。

generalization-rule の例として、述語と上位概念に一般化する規則の記述を図8に示す。「睡眠薬を飲む」を「自殺」に一般化する case-by-case-rule を図9に示す。概念の上下関係の知識(グラフ)もまた H-net で記述されている。概念の上下関係には ISA の関係(クラスとサブクラスの関係)と Class-Instance の関係の2種類がある。図10a)のような外部表現で概念の上下関係を与えると図10b)のような H-net 表現に変換されて蓄えられる。

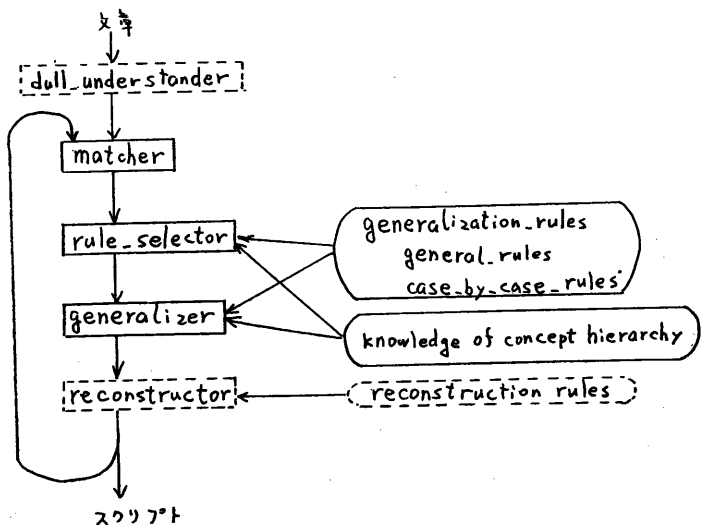


図7 SEM 1.5 の構成

aufheben_pred

```

| intension
(lambda ((p . xv) (xg . xw))
  (parents xp *pp)
  (parents xg *gg)
  (conj1 *pp *gg *r)
  (aufheben_vars xv xw xv *types)
  (hreturn `((,xr ,xw) ,*types)))

```

図8 general_rule の例

飲む_rule

```

| intension
(lambda ((飲む xx 睡眠者) (xg . xw))
  (parents xg *gg)
  (or (eg xg '自殺) (memg '自殺 *gg))
  (aufheben_vars (xx 睡眠者) xw xv *types)
  (hreturn `((自殺 ,xw) ,*types)))

```

図9 case-by-case_rule の例

generalizer は一般化の規則を HNR の関数 deduce によ

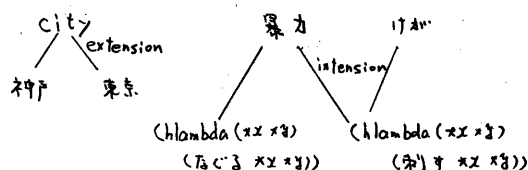
```

(ins 神戸 city)
(ins 東京 city)
(isa 花子 殺人)
(isa 刺す 殺人)
(isa 刺す けが)

```

り、(deduce((aufheben_pred (刺す 花子)))) に

(a) 外部表現



(b) 内部表現

図10 概念の上下関係の表現

り、(暴力 xx xg) が得られる。generalizer は定数と変数へ置きかえのりストをも、といて、同じ定数は同じ変数に置きかえられ。deduce は Prolog と同じように top-down, depth-first の推論を行い失敗したバックトラッキングする。

SAM は HNR により容易に実現される。スクリプトを deduce するだけでよい。実際のシステムでは、図2に示した *text という変数を用いず、1つの文章のイベント・ストラクチャ中の各述語を一時的に外延として付加しておいて、スクリプトを deduce するという方法をとっている。たとえば、イベント・ストラクチャ中に (刺す 太郎 次郎) という述語があったら、(太郎 次郎) を「刺す」の外延として一時的に宣言して、スクリプトを演繹すれば、スクリプト中の (刺す xx xg) や (暴力 xx xg) が成功し、変数がバインドされる。こうして、スクリプト中のすべての述語の演繹に成功すればスクリプトにあてはまる実例であり、1つでも失敗すればスクリプトにあてはまらない文章であるとの判断がなされる。

3. 実験と評価

スクリプトの帰納的学習の有効性を評価するための実験を行、た。実験対象としては、日本経済新聞の1月から5月までの新聞記事で、殺人、自殺、心中、火事、乾落事故に関するもの42件を用いた。はじめ、これらの記事を次の6つのグループに分類する。(a)殺人、(b)自殺、(c)心中、(d)殺人、(e)焼死、(f)乾落死。この分類は、キーワードだけでは選別できないようなお互いに接近した微妙な分類であり、筆者が白分でもは、きりした基準をもたずに、それぞれの文章を読んで直感的に分類を行った。これは、SEM がこの直感的分類と一致した分類を行えるようなスクリプトを出力できるかどうかの評価しようとするためである。表1に、使用した記事の見出しと参考のため示す(実験では本文だけを用い、見出しは用いていない。)

SEM の出力したスクリプトを用いた SAM による分類の実験結果を図11に

case-by-case の面を与えた場合である。3)は訓練例を与える順序と教習がコントロールした(典型的と思われる例から順に与えた)場合であり、1)はランダムに与えた場合である。

出力されたスクリプトの例としては、ii)3)の場合、訓練回数5で出力されたスクリプトは (boryoku *v1 *v2)(jisatsukonkei *v3 *v4)(ishop *v5 *v6)(korosu *v7 *v8) であった。ii), iii) で与えたシソーラスには、図 10 a) のような関係が 114 組入っており、iii) で与えた case-by-case-rules の数は 5 つである。同じ知識と規則を用いて (b) グループのスクリプトを学習して分類する実験も行、たが、ほぼ同様の結果が得られた(すなわち、(a) グループ専用で微調整した知識とルールではない)。(b) グループのスクリプトとしては、(ciku *v1 *v2)(shi *v3 *v4)(kurushimi *v5 *v6) などが出力された。SEM の 1 回の訓練は CPU time で約 10 秒、SAM の 1 つの文章の判断は CPU time で約 1 秒かかる。

図 11 をみると、訓練例を与える順序をランダムにすると、スクリプトが一般化されすぎる事がわかる。シソーラスや case-by-case rules と与えよとて結果が悪くなる場合もあるのは、SAM のほうで、1 つの概念に該当する下位概念がふえたために推測が成功する率が上がるからである。図 11 の iii) 3) とみると、ほぼ満足できる学習が行えたといえる。しかし、上述した出力スクリプトで、本意は自殺と殺人を行った *v3 と *v7 が等しいはずだが、dull-understander が照応の取扱いと行、ていなため、これは検出されてない。

4. 考察

自然言語理解システムの用いる知識の学習システムの必要性を主張し、スクリプトを文章から帰納的に学習する手法と実験システムを与え、実験による有効性を示した。本論文で用いた学習の手法自体は、人工知能の分野では特に新しいものではないが、汎用性と有用性への配慮に配慮した実験システムを作成し、実際に、自然言語理解システムの用いる知識の学習の可能性を示したことは有意義であると考える。今後、もっと複雑なスクリプトも学習できるシステムを構築するためには、SAM から SEM へのフィードバック、スクリプトの評価基準の導入、評価基準に基づいて一般化を制御すること、disjunction をスクリプト中に含めること、照応を取り扱う規則とその学習などを考えておかなければならないであろう。

参考文献

- 1) 猪瀬希彦: シリオを用いた論文抄録理解・作成システム。
情報処理学会論文誌, Vol. 24, No. 1 (1983)
- 2) Shapiro, E.Y.: Inductive Inference of Theories from Facts, Research Report, No. 192, Yale Univ. (1981).
- 3) Winston, P.H.: Learning and Reasoning by Analogy, Comm. ACM, Vol. 13, No. 12, pp. 689-703 (1980).
- 4) Buchanan, B.G. et al.: Dendral and Meta-Dendral: Their Application Dimension, A.I., Vol. 11, pp. 5-24 (1978).
- 5) Langley, P.: Data-Driven Discovery of Physical Laws, Cognitive Science, Vol. 5, pp. 31-54 (1981).
- 6) Feigenbaum, E.A. et al. (eds): The Handbook of Artificial Intelligence, Chapter 14, pp. 323-511, Pitman (1981).
- 7) Shank, R.C. et al.: Scripts, Plans, Goals and Understanding, John Wiley and Sons (1977).
- 8) Shank, R.C. et al.: Inside Computer Understanding, Lawrence Erlbaum Associates (1981).
- 9) Shank, R.C.: Language and Memory, Cognitive Science, Vol. 4, pp. 243-284 (1980).
- 10) 池本隆, 猪瀬: 帰納的推論の記述に有用な知識表現の一種。
情報処理学会論文誌, Vol. 24, No. 1 (1983)