

医療診断用音声翻訳システム MedSLT における日本語規則の作成

中尾雪絵 (大阪大学),

Manny Rayner (ICSI/NASA), Beth Ann Hockey (UCSC/NASA),

Pierrette Bouillon, Nikos Chatzichrisafis (University of Geneva),

神崎享子, 井佐原均 (独立行政法人情報通信研究機構)

1. はじめに

医師と患者の間に言語障壁がある場合、医療診断を行うことは大変困難である。この問題を解決するために、われわれは医療診断のドメインでの音声翻訳システム MedSLT を開発している。

医療診断という限られたドメインを設定することは、音声翻訳システムを構築する際、有利な点がある。医師と患者の相互のやりとりはある程度制約されている。特に、医師が患者に聞く主要な質問のパターンは、比較的決まっており、数も多くない。また、医師が患者に症状を質問しながら診断していく関係上、医師主導の質問形式である。そのため、患者側は、限られた言葉、「はい/いいえ」や「左/右」、数、あるいは具合の悪い場所を指で指示するなどで答えればすむ。このことから、医療診断は、片方向の音声翻訳システムが有効なドメインであると考えられる。

MedSLT (MedSLT, 2005) は一般的なプラットフォームを用いたオープンソースのプロジェクトである。初期のバージョンについては、(Rayner and Bouillon, 2002; Rayner et al., 2003)で述べられている。文法は、単一化文法形式で書かれており、文法ベース言語モデル (grammar-based language models, GLMs) とオープンソース Regulus toolkit (Regulus, 2005) を使った生成器 (ジェネレーター) にコンパイルされる。言語モデルは GSL (Grammar Specification Language) 形式で、

Nuance プラットフォームで使用するのに適している。また、翻訳は中間言語方式で行われる。

この翻訳のための中間言語に対して、われわれは、日本語の翻訳ができるように日本語規則を作成している。

ここで採用したアプローチの特色の一つは、一方向制限入力という制約はあるが、高速かつ安価にシステムを開発し、多言語化できることにある。世界を多様な国の人々が行き交う昨今、例えば災害などの非常事態における自然言語処理技術の貢献を考えた場合に、限られたドメインであっても、「高速かつ安価」というアプローチは、有効であると考えられる。

2. MedSLT のしくみ

MedSLT は基本的には、定型句や定型文ベースの翻訳と、ルールベースの翻訳の中間的なものである。実行中、システムは、入力言語の表現のバリエーションを許す柔軟な定型表現翻訳を行い、通常 of 定型表現翻訳に近い。しかし、蓄積する定型表現数を最小に抑えるために、単一化文法からこれらの定型表現のパターンが生成されるようにしており、これは、オープンソースである Regulus (Regulus, 2005) というプラットフォームを使って実行される。

定型表現と文法的なルールの両者を使うと、信頼性が高く、実行スピードが速く、さらに拡張や修正などを簡単に行えるというメリットがある。

翻訳モジュールは、SICStus Prolog で実装され、中間言語がベースになっている。翻訳は、

四段階で行われる。

- 1) 原言語の表現から中間言語へのマッピング
- 2) 省略などの処理
- 3) 中間言語から目標言語の表現へのマッピング
- 4) コンパイルした Regulus Grammar を用いて、目標言語の生成を行う。

省略などの処理に関して、原言語の認識の段階で作られた意味表現は文脈処理モジュールに送られ、前の対話の文脈にその意味表現をおいて翻訳する。たとえば、“Does the pain occur every day?”という質問が前にあれば、次の質問の“Several times a day?”が、“Does the pain occur several times a day?”と等価な表現として翻訳される。

中間言語の表現法は基本的には、各節をフラットな「キー・値」ペアのリストとして扱うことである。たとえば、“is the pain better when you lie down?”の表現を中間言語で表現すると、以下ようになる。

```
[[utterance_type,ynq],[symptom,pain],  
[event,become_better],[tense,present],  
[voice,active],[sc,when],  
[clause,[[utterance_type,dcl],  
[pronoun,you],[action,lie_down],  
[tense,present],[voice,active]]]]
```

中間言語の表現は原言語に訳し戻され、ユーザーに提示される。ユーザーは、システムが訳に失敗していると思えば、続く処理を中止する。ユーザーが訳し戻されたものを正しいと承認すると、中間言語の表現は、目標言語の表現に翻訳される。つまり、生成用の文法を使って、目標言語の表層的な単語の並びへ翻訳され、音声合成へと受け渡される。

また、ユーザーはヘルプモジュールを任意で

使用することができる。ヘルプモジュールを設定すると、ユーザーが発話した際に、発話内容を認識器（リコグナイザー）が認識するしないに関わらず、その発話に似た複数の文候補がヘルプ画面に表示されるようになる。この文候補は、ヘルプコーパスに集められた膨大な例文群の中から選ばれる。その際、ユーザーの発話を統計的に処理したときに抽出される n グラムと同じ数の n グラムを持つ文が選ばれるようになっている。

ヘルプモジュールを使うと、システムに慣れていないユーザーも、システムの適用範囲を自然に学習できる。また、システムが認識を誤った場合でも、認識させようとした文の類似候補を、有効にユーザーに提示することができる。

3 . 日本語規則の設計

MedSLT には認識用と生成用の 2 つの文法がある。認識用の文法には、56 のルールと、868 語のエントリーがある。生成用には、54 のルールと、850 語のエントリーがある。ほとんどの規則と語彙項目は 2 つの文法で重複している。意味表現は、中間言語と同様で、節が「キー・値」のペアのフラットなリストになっている。以下の例は、「横になると痛みはおさまりますか？」の意味表現である。

```
[[utterance_type,sentence],  
[symptom,itami],[event,osamaru],  
[tense,present],[postpos,causal],  
[clause,[[utterance_type,dcl],  
[cause,yoko],[path_proc,naru],  
[postpos,ni],[tense,present]]]]
```

文法の基本構造

NP, PP, VP, 後置詞の並びといったレベルで、基本的な節のパターンを分類すると 6 パターンほどある。たとえば、以下のように記述している。

CLAUSE → PP NP ga VP

このようなルールは、認識用文法と生成用文法で、それぞれパラメーター化される。たとえば、認識用では、「が」と「は」は変えることができるが、生成用では、適切な「が」と「は」のどちらかを選択しなければならない。

また、VP のパターンを分類すると、さらに6パターンある。例えば、以下のように記述している。

VP → NP wo VBAR: [subcat=trans]

そのほか、前置修飾の PP の節を許すルールや、NP の構造化のルール、また、接続助詞を伴う従属節から PP を形成するルールなどがある。日本語の統語的派生の例として、「横になると痛みは治まりますか？」の日本語文の構造をやや簡易化して書くと以下ようになる。

```
sigma
  question
    pp
      subordinate_clause
        vp
          pp
            np lex(yoko)
            p lex(ni)
            vbar
              v lex(naru)
            sc lex(to)
            np lex(itami)
            lex(wa)
            vp
              vbar
                v lex(osamari) lex(masu)
            lex(ka)
```

4 . 認識

われわれは認識用の文法を Nuance リコグナイザーへコンパイルするために、Regulus のパッケージを使用した。単一化文法で、56の規則と868の語彙項目は、文脈自由文法にコンパイルされ、1561の規則となった。コンパイルに要する時間は、3.2GHzのラップ

トップ(P4 processor)で6分30秒かった。

5 . 中間言語からの変換

中間言語は、基本的には英語に基づいている。しかし、たとえば、原因を表す節は標準的な表現に変換されているので、原因は、常に、動詞の構造で表される。たとえば、“Does chocolate give you headaches?” は、“Do you get headaches when you eat chocolate?”と等価である。つまり「チョコレート」が主語に立つことができる英語での入力に対して、「チョコレートを食べて」とのように、原因を述語動詞の構造で表現しておけば、日本語への翻訳が容易になる。これは、英語と日本語の間の翻訳を考える場合に有効なルールである。

句は、キーと値のペアのリストで表現されており、それらの記述の順番は重要ではない。翻訳ルールはほとんどの場合、キーと値のペアからキーと値のペアへと変換される。以下に変換の例を示す。

[body_part, jaw] -> [body_part, ago]

[adj, dull] -> [degree, nibuku]

[freq, often] -> [frequency, yoku]

中間言語の表現と日本語の表現が異なる場合には、キーと値のペアのリストを、英語と日本語の間でマッピングする必要がある。たとえば、日本語の「疲れる」は、「experience」と「fatigue」にマッピングされ、「undo」「wo」「suru」は「on_exertion」にマッピングされる。その例を以下に示す。

[[loc, on_exertion]] -> [[activity, undo],

[path_proc, suru], [postpos, causal]]

[[state, experience], [secondary_symptom, fatigue]] -> [[path_proc, tsukareru]]

6 . 評価

われわれは、12人のネイティブの英語話者からデータを集めた。具体的には、彼らに医師を演じ

てもらい、患者役のわれわれに頭痛の症状について質問をしてもらった。「患者」は、頭痛の病名リストの中から、あらかじめ一つの病名と、その病気に伴う症状を覚えておき、医師の質問に yes/no やジェスチャーで答える。「医師」が病名を正しく特定することがタスクである。集まった 870 発話に対し、英語の文法をカバーしているかどうかを調べた。

最初に、英語話者が、発話によって得られた訳し戻しを評価した。以下に、評価の結果を示す。評価は、ワードエラーの割合(WER)、文エラーの割合(SER)、意味エラーの割合(SemER)の点から行われた。

ワードエラーは、発話に対して、認識された単語の中から、誤っている語、新たに加えられた語、削除された語の割合を算出して得られる。文エラーとは、認識された文のうち、認識を誤った文の割合である。意味エラーとは、発話と認識文が異なっている文のうち、意味合いも異なっているものの割合を算出したものである。

また、われわれのシステムで取り扱っている発話、取り扱っていない発話、全てのデータを対象にそれぞれ割合を出した。

	Files	WER	SER	SemER
システムの範囲内の発話	417	5.7%	19.4%	18.5%
範囲外の発話	453	57.5%	99.8%	87.9%
合計	870	34.3%	62.0%	54.6%

表 1 実験の評価結果

その結果、取り扱っている発話は、約 80% が正しく認識されたが、取り扱っていない発話はエラー率が高くなった。これは、取り扱う発話をユーザーに示すヘルプモジュールの役割が大きいことを示す。実際、発話が正しく認識されない率を、実験の最初と最後で比較してみた。その結果、ヘルプモジュールを使った場合、17.2%の減少となったのに対して、ヘルプモジュールを使わない場合は、6.5%減っただけだった。

一方、日本語の出力の評価実験結果については、認識できた発話のうち、80.2%の日本語訳が「良い」と評価され、「まあ良い」と評価されたものが 7.9%あった。訳が抽出されなかったものは 10.7%で、訳が正しくないと評価されたものはわずか 1.1%にとどまった。この評価は多数決制をとっており、3人のネイティブの日本語話者によって行われた。

7. まとめ

医療診断用音声翻訳システムは、オープンソースを利用し実装しているシステムで、言語翻訳は、中間言語を介して目標言語に翻訳するしくみである。これまでのところ、われわれは英語からフランス語、日本語、フィンランド語への翻訳システムの開発を進めている。また日本語からフランス語への翻訳についても開発を開始した。すでに試作品ができあがりつつあり、その結果については別に発表の機会を持ちたい。

参考文献

- MedSLT, 2005. <http://sf.net/projects/medslt/>, as of 8 January 2005.
- M. Rayner and P. Bouillon. 2002. "A phrasebook style medical speech translator", In *Proc. of the 40th Annual Meeting of the ACL (demo track)*, Philadelphia, PA.
- M. Rayner, B.A. Hockey, and J. Dowding. 2003. "An open source environment for compiling typed unification grammars into speech recognisers." In *Proc. of the 10th EAACL (demo track)*, Budapest, Hungary.
- Regulus, 2005, <http://sf.net/projects/regulus/>, as of January 2005.
- Nuance, 2005. <http://www.nuance.com>, as of January 2005.