

意味主導の日本語構文解析手法の 確立を目指した基礎研究

東北大学 工学部 電気情報物理工学科
通信コース
大堀研究室 4年

B8TB2211 牧野雅紘
指導教員：大堀 淳 教授
菊池 健太郎 助教

研究背景と目的

西洋語での構文解析

主語・述語関係を核とした文法規則を定義し
文法主導で構文解析を行う

日本語でも同じ手法で構文解析されることが多い

日本語の特徴

- 主語・述語関係がない
- 語順の曖昧性

文法主導の構文解析
では**不十分**と考える

研究背景と目的

本研究の最終目標

日本語の**意味主導**の構文解析

意味主導の構文解析とは？

- 文法規則をほとんど必要としない解析
- 語順が変わっても高い精度で解析
- 三上文法（後述）を取り入れた解析

本研究では大堀(意味主導の日本語解析について)による提唱を参考に行う

日本語話者が自然に行っている意味に従った構文解析が可能

三上文法について

三上文法とは？

言語学者の三上章が体系化した日本語文法

三上文法の概要

- 主語は**存在しない**
- 助詞の"は"が**提題**を表す
- 助詞の"は"が他の助詞"**がのにを**"を代行する
- 配置される**連用修飾語の順番は自由**

三上文法について

主語は存在しない

○主語とは？
文中の**主格**
述語と呼応する性質がある

“Xは”と“Xが”のXは主語ではない

メバルは煮る ➡ メバルは動作主ではない

私は紹介する
私が紹介する
彼女を紹介する

述語と呼応していない

三上文法について

提題を行う“は”

- 提題とは？
伝える内容の**主題を挙げる**こと
- “は”は主格ではなく提題を行う助詞

例文：象は鼻が長い

主題（話すトピック）：象

主格（動作主）：鼻

述語（動作、性質）：長い

三上文法について

“がのにを”を代行する“は”

象は(が)大きい

象は(の)鼻が長い

象は(に)話しかけても話さない

象は(を)みんなが好き

三上文法について

連用修飾語の語順は自由

主格(が)、与格(に)、対格(を)の
語順は自由

例:(わたしが)(あなたに)(数学を)教える

→()の語順は自由

システム構築戦略

日本語解析に必要なこと

- は格の文における機能の同定
- 語順等の構文論に依存しない解析手法

システム構築戦略

1. 意味表現の定義

例：“象は速く走る”の意味表現のレコード

```
{  
  種類 = 動詞文,  
  提題 = {名詞 = “象”},  
  述語 = {動詞 = “速く”, 主格 = {名詞 = “象”}}  
}
```

システム構築戦略

2. 文節ごとの意味表現

例：“象は”の考えられ得る意味表現
のレコードのリスト

```
[{
  種類 = 動詞文,
  提題 = {名詞 = “象”},
  述語 = {主格 = {名詞 = “象 ”}}
},
{
  種類 = 動詞文,
  提題 = {名詞 = “象”},
  述語 = {主格 = {の格 = “象”}}
}, . . .
]
```

システム構築戦略

3. 自然結合演算

例：“象は”と“走る”の部分意味表現の
自然結合演算

<“象は”の部分意味表現>

```
[{
  種類 = 形容詞文,
  提題 = {名詞 = “象”},
  述語 = {主格 = {の格 = “鼻”}}
},
{
  種類 = 動詞文,
  提題 = {名詞 = “象”},
  述語 = {主格 = {名詞 = “象”}}
}]
```

<“走る”の部分意味表現>

```
[{
  種類 = 動詞文,
  述語 = {動詞 = “走る”}
}]
```



2つのレコードのリストを自然結合演算
(それぞれのレコードごとに共通の属性が一致しているかをテストし、一致していればレコードの和をとる)

<“象は走る”の意味表現>

```
[{  
  種類 = 動詞文,  
  提題 = {名詞 = “象”},  
  述語 = {動詞 = “走る”, 主格 = {名詞 = “象”}}  
}]
```

```
{  
  種類 = 動詞文,  
  述語 = {動詞 = “走る”}  
}
```

```
{  
  種類 = 動詞文,  
  提題 = {名詞 = “象”},  
  述語 = {主格 = {名詞 = “象”}}  
}
```



システム構築戦略

象は
の部分意味表現

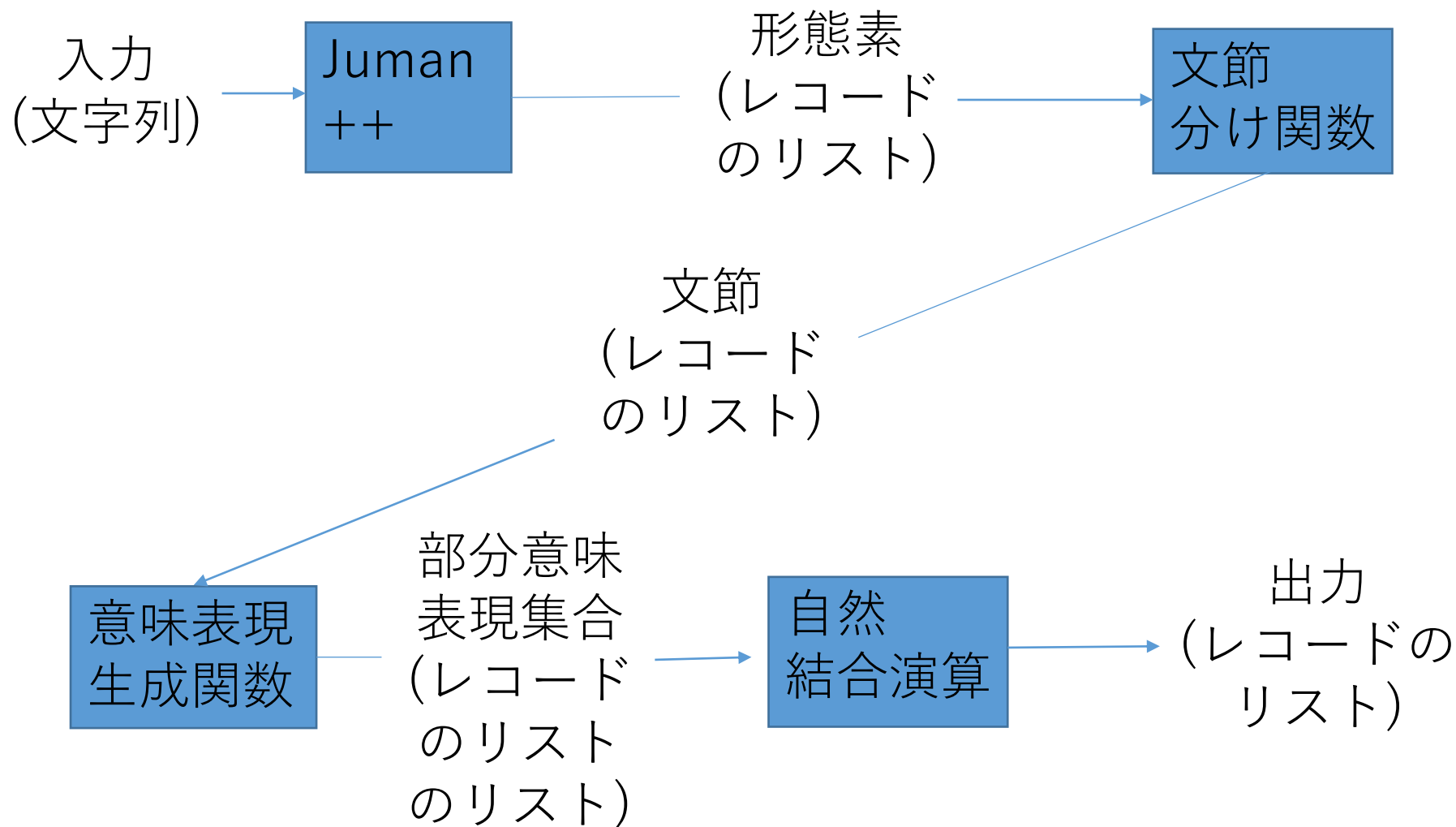
自然結合演算

速く
の部分意味表現

象は速く走る
の意味表現の候補

走る
の部分意味表現

システム構築戦略



システム構築～Juman++～

```
type jumanOutputTy =  
  {  
    表層形 : string,  
    読み : string,  
    見出し語 : string,  
    品詞大分類 : string,  
    品詞大分類_ID : int option,  
    品詞細分類 : string,  
    品詞細分類_ID : int option,  
    活用型 : string option,  
    活用型_ID : int option,  
    活用形 : string option,  
    活用形_ID : int option,  
    意味情報 : jumanOutputSemTy option  
  }
```

形態素ごとの情報を以下
のようなレコードで保持

システム構築～文節分け関数～

繰り返しの終了

```
fun separate_record lis =  
  if lis = nil then  
    case #4_品詞大分類 record of  
      "形容詞" => [Type.形容詞 {形容詞 = #1_表層形 record, 形容詞情報 = record}]  
    | "動詞" => [Type.動詞 {動詞 = #1_表層形 record, 動詞情報 = record}]  
      .  
      .  
  else
```

名詞+助詞で分ける

```
    if #4_品詞大分類 (List.hd lis) = "助詞" then  
      case #1_表層形 (List.hd lis) of  
        "は" => List.@ ([Type.名詞は {名詞 = #1_表層形 record, 名詞情報 = record, 助詞情報 = List.hd lis}], separate (List.hd (List.tl lis))  
        (List.tl (List.tl lis)))  
      | "が" => List.@ ([Type.名詞が {名詞 = #1_表層形 record, 名詞情報 = record, 助詞情報 = List.hd lis}], separate (List.hd (List.tl lis))  
        (List.tl (List.tl lis)))  
      | "の" => List.@ ([Type.名詞の {名詞 = #1_表層形 record, 名詞情報 = record, 助詞情報 = List.hd lis}], separate (List.hd (List.tl lis))  
        .  
        .  
    else
```

```
      case #4_品詞大分類 record of  
        "形容詞" => List.@ ([Type.形容詞 {形容詞 = #1_表層形 record, 形容詞情報 = record}], separate (List.hd lis) (List.tl lis))  
      | "動詞" => List.@ ([Type.動詞 {動詞 = #1_表層形 record, 動詞情報 = record}], separate (List.hd lis) (List.tl lis))  
      | "名詞" => List.@ ([Type.名詞 {名詞 = #1_表層形 record, 名詞情報 = record}], separate (List.hd lis) (List.tl lis))  
      | _ => raise Fail "unknown hinshi"
```

助詞以外を分ける

システム構築～意味表現生成関数～

```
fun undone t =  
  case t of  
    Type.名詞は s =>  
      [  
        Dynamic.dynamic  
          {  
            種類 = "形容詞文",  
            述語 = {主格 = {名詞 = #名詞 s}},  
            提題 = {名詞 = #名詞 s}  
          },  
        Dynamic.dynamic  
          {  
            種類 = "形容詞文",  
            述語 = {主格 = {の格 = #名詞 s}},  
            提題 = {名詞 = #名詞 s}  
          },  
        .,  
        .,  
        Dynamic.dynamic  
          {  
            種類 = "動詞文",  
            述語 = {主格 = {名詞 = #名詞 s}},  
            提題 = {名詞 = #名詞 s}  
          },  
        .,  
        .  
      ]
```

```
| Type.名詞が s =>  
  [  
    Dynamic.dynamic  
      {  
        種類 = "形容詞文",  
        述語 = {主格 = {名詞 = #名詞 s}}  
      },  
    .,  
    .  
  ]  
  
| Type.動詞 s =>  
  [  
    Dynamic.dynamic  
      {  
        種類 = "動詞文",  
        述語 = {動詞 = #動詞 s}  
      }  
    ]  
    .  
    .
```

文節ごとに取り得る
部分意味表現を生成

システム構築～自然結合演算～

```
fun listjoin (lis:Dynamic.dynamic list list) =  
  case lis of  
    nil => raise Fail "nil to listJoin"  
  | h :: nil => h  
  | x :: y :: t => listjoin (_join(x, y) :: t)
```

文節ごとの
部分意味表現を自然結合演算

出力例(“象は鼻が長い”)

```
{  
  "提題" : {"名詞" : "象"},  
  "種類" : "形容詞文",  
  "述語" :  
    {"主格" : {"の格" : "象", "名詞" : "鼻"},  
     "形容詞" : "長い"}  
}
```

象の鼻が長い

```
{  
  "提題" : {"名詞" : "象"},  
  "種類" : "動詞文",  
  "述語" :  
    {"主格" : {"の格" : "象", "名詞" : "鼻"},  
     "連用修飾" : "長い"}  
}  
{  
  "提題" : {"名詞" : "象"},  
  "種類" : "動詞文",  
  "述語" :  
    {"に格" : "象", "主格" : {"名詞" : "鼻"},  
     "連用修飾" : "長い"}  
}  
{  
  "提題" : {"名詞" : "象"},  
  "種類" : "動詞文",  
  "述語" :  
    {"を格" : "象", "主格" : {"名詞" : "鼻"},  
     "連用修飾" : "長い"}  
}
```

述語動詞が存在していないのに動詞文
が候補として挙がっている

出力例(“私は彼に仕事を勧める”)

```
{  
  "提題" : {"名詞" : "私"},  
  "種類" : "動詞文",  
  "述語" :  
    {  
      "に格" : "彼",  
      "を格" : "仕事",  
      "主格" : {"名詞" : "私"},  
      "動詞" : "勧める"  
    }  
}
```

私が彼に仕事を勧める

```
{  
  "提題" : {"名詞" : "私"},  
  "種類" : "動詞文",  
  "述語" :  
    {  
      "に格" : "彼",  
      "を格" : "仕事",  
      "主格" : {"の格" : "私"},  
      "動詞" : "勧める"  
    }  
}
```

の格の対象となる名詞
が存在していない

出力例(“彼に私は仕事を勧める”)

```
{  
  "種類" : "動詞文",  
  "述語" :  
    {  
      "に格" : "彼",  
      "を格" : "仕事",  
      "主格" : {"名詞" : "私"},  
      "動詞" : "勧める"  
    }  
}
```

彼に私が仕事を勧める

出力例(“メバルは煮る”)

```
{  
  "提題": {"名詞": "メバル"},  
  "種類": "動詞文",  
  "述語": {"を格": "メバル", "動詞": "煮る"}  
}
```

メバルを煮る

```
{  
  "提題": {"名詞": "メバル"},  
  "種類": "動詞文",  
  "述語": {"主格": {"名詞": "メバル"}, "動詞": "煮る"}  
}
```

メバルが煮る

```
{  
  "提題": {"名詞": "メバル"},  
  "種類": "動詞文",  
  "述語": {"に格": "メバル", "動詞": "煮る"}  
}
```

メバルに煮る

```
{  
  "提題": {"名詞": "メバル"},  
  "種類": "動詞文",  
  "述語": {"主格": {"の格": "メバル"}, "動詞": "煮る"}  
}
```

の格の対象となる名詞
が存在していない

現状のまとめ

- 文中に存在する**“は”の機能の候補**を複数挙げる
ことが可能となった
- **語順に依存しない構文解析**が可能となった

今後の展望

プログラムの改良

- 動詞の存在しない動詞文の候補を排除
- 形容詞の存在しない形容詞文の候補を排除
- 修飾対象の用言がないにも関わらず連用修飾語が存在する候補文を排除
- 正しい構文解析結果のみを出力

システムの応用

- 美しい日本語に訂正してくれるシステムへの応用

ご清聴ありがとうございました.