

動的な教材開発を目指して（1）：ドイツ語の冠詞 と名詞を解析する

栗山, 暢

九州大学大学院言語文化研究院言語環境学部門：准教授：言語教育学

<https://doi.org/10.15017/18355>

出版情報：言語文化論究. 25, pp.45-52, 2010-03. 九州大学大学院言語文化研究院
バージョン：
権利関係：

動的な教材開発を目指して (1)

——ドイツ語の冠詞と名詞を解析する——

栗 山 暢

コンピュータを利用した外国語学習教材はあまたあるが、そのほとんどが、選択肢の中から正解を選ばせるような問題、さもなければ単語1語といったごく単純な入力に対しそれが正解と完全に一致した時のみ正答と認めるといった問題を提供するものようである。そうした問題を提供することで得られる学習効果を否定するつもりはもとよりのないが、学習効果という点からも、そしてまた教育におけるコンピュータ利用の可能性という点からも、そのような状態は理想的とはいいがたい。筆者は、学習者の書いた誤りを含むドイツ語文を解析し、正解に向けてふさわしいコメントを返すようなコンピュータプログラムの開発を目指している。本稿は、そのもっとも基礎的な、冠詞類と名詞の組み合わせを解析する処理を解説するものである。

1. 捕捉すべき誤り

冠詞類・名詞それぞれについて、初学者が往々にして書きがちな誤った形を含むあらゆる変化形をあらかじめ辞書に登録しておく。辞書に登録されていない形が解答の一部として入力された時にはその時点で以後の処理を拒否することになるので、辞書はなるべく大きなものであるのが望ましい。これは、逆に言うと、タイプミスによるものと思われる明らかにおかしい入力を拒否することでもあるので、学習者の注意を本来の学習に集中させるといった効果を持つことにもなる。

分析すべき誤りは、冠詞類の形にかかわる誤り、名詞の形にかかわる誤り、名詞句の性・数・格に関する誤りの三種である。

冠詞類の形に関して捕捉すべき誤りは筆者の経験によれば以下の通り。

article1: dieserer 型

article2: diesie 型

article3: einie 型

article4: unsen 型

article5: meiner 型

article1 は、定冠詞類の末尾の”er”を語尾と思わずに、その後に語尾をつけてしまう誤り。jeneres, jederem など。article2 は、定冠詞類の女性（単数）・複数それぞれ1、4格の語尾”e”を、定冠詞の語尾の連想から”ie”とし、また、中性（単数）1、4格の語尾を同様に”as”とする誤り。diesas など。article3 は不定冠詞（類）の女性・複数それぞれ1、4格および中性（単数）

1、4格での同様の誤り。meinie, deinas など。article4 は不定冠詞類 unser、euer について、語末の ”er” を語尾だと思い、それ自体を変化させるもの。unsem、eues など。article5 は、不定冠詞（類）において語尾の欠ける男性（単数）1格、中性（単数）1、4格に語尾をつける誤り。seiner Vater、meines Haus など。

名詞の形に関して捕捉すべき誤りは以下の通り。

- noun1: des Mann 型**
- noun2: des Arzts 型**
- noun3: des Vateres 型**
- noun4: der Fraus 型**
- noun5: der Frauens 型**
- noun6: des Fraus 型**
- noun7: des Students 型**
- noun8: dem Student 型**
- noun9: des Studentens 型**
- noun10: Nachbaren 型**
- noun11: den Kinder 型**
- noun12: den Kinderen 型**
- noun13: den Frauenn 型**
- noun14: die Vater 型**
- noun15: die Kindes 型**
- noun16: den Kindn 型**
- noun17: den Mannsn 型**
- noun18: des Geheimnisses 型**
- noun19: die Geheimnisse 型**
- noun20: die Freundinnen 型**
- noun21: eine Leute 型**
- noun22: meiner Leute 型**

noun1 は、男性（単数）・中性（単数）2格で語尾 ”(e) s” をつけ忘れる誤り。noun2 は同様の場合に必要な口調上の ”e” をつけ忘れる誤り、noun3 はあってはならない口調上の ”e” をつけてしまう誤り。noun4 は女性（単数）2格に、noun5 は複数2格に ”(e) s” をつける誤り。noun6 は、女性名詞を男性ないし中性名詞だと思う誤り（単数2格の時のみ）。

noun7 から noun10 までは男性弱変化名詞に関するもので、noun7 は単数2格に ”(e) s” をつけるもの、noun8 は3格、4格に ”(e) n” をつけない誤り、noun9 は弱変化させた単数2格にさらに ”(e) s” をつけるもの、noun10 は ”en” でなければならない語尾を ”n” と書く、およびその反対の場合である。

noun11 から 13 は複数3格に関するもので、noun11 は必要な ”n” を加えないもの、noun12 は誤って口調上の ”e” を入れるもの、noun13 は不必要な ”n” を最後につけてしまうもの。

noun14は単数形をそのまま複数形として使っている場合、noun15は”(e) s”をつけて複数形にしたつもりでいる場合。noun16は複数形を導かないまま複数3格のつもりで”n”をつけ、noun17は”(e) s”をつけた誤った複数形に3格”n”をつけたもの(この、たとえば”denMannsn”というのはほとんどあり得そうもないけれど、noun16の”den Kindn”などは実際に目にする事は少なくない)。

noun18からnoun20は特定の形の名詞にかかわるもので、noun18と19は”nis”に終わる名詞の単数2格および複数形、noun20は”in”と終わる女性名詞の複数形にかかわるもの。

noun21は、単数形を持たない名詞を単数扱いにする誤り。

noun22はやや複雑な判断をするもので、たとえば”meiner Leute”という入力通常では複数2格としか解しえないが、学習者が”Leute”という単語を女性(単数)だと思っていることがしばしばあると思われるので、そのような場合に対応する。すなわち、”Leute”を2格で書かなければならない場合にはそのまま正解とし、3格で書くことが求められている場合には、「格が間違っている」というメッセージではなく、「Leuteは女性名詞ではない(複数形である)」というメッセージを発することになる。3格で書かなければならないところを2格で書いてしまう誤りよりは、単数形を持たない名詞を女性名詞だと思って単数扱いする(”meiner Leute”は女性・単数・3格のつもりである)誤りの方がよりありそうに思われるからである(この判断は今のところ経験に基づいているのみであるから、システムが実際に稼働するようになった時点で調査する必要があるだろう)。

冠詞類と名詞との組み合わせに関する誤りは以下の通り。(info:Mann:der:s(pl):2は、”Mann”という名詞を単数(s)(ないし複数(p))2格で定冠詞をつけて書くことを求めているものとする。

np1: 性・数・格を決定できないもの。”das Mann”など。

np2: 格を間違えているもの。info:Mann:s:3に対して”den Mann”など。

np3: 数を間違えているもの。info:Mann:p:4に対して”den Mann”など。

np4: 格、あるいは数のいずれかを間違えているもの。

np5: 冠詞類の種類を間違えているもの。”dieser”を使うべきところで”der Mann”など。

np6: 冠詞を使うべきところであるのに無冠詞のもの。

np7: 無冠詞の指示であるのに冠詞類を使っているもの。

このうちnp4がやや曖昧であるが、これは、”Lehrer”など、単複同形の名詞を、単数・2格、ないしは複数・1格(4格)で書くことを求めた時に”der Lehrer”と入力があった場合に限られることが期待される。そうだとすると、単数・1格、複数・2格のどちらも正しい形なのであるから、解答者がどのどちらのつもりであるか判断できないのもやむを得ない。

2. 実装

使用する言語はRuby (version 1.9.1)。

用意するクラスは、全体をまとめるものとしてBeckmesser、その下にDictionary、QuestionInfo、Answer、Parser、MessageMaker、このうち最後のひとつを除いたもの下にはそれぞれいくつかのクラスがある。

動作のイメージは、

```

info = "Mann:dieser:p:3"
ans = "jeneren Männer"
bm = Beckmesser.run(info, ans)
bm.message # => "0" => ["article1"], "1" => ["noun11", "np5"]

```

といったところである。0番目の単語（"jeneren"）について上記の "article1" の誤り、1番目の単語（"Männer"）についてふたつの誤りを指摘している。

Dictionary クラスにおいては、**QuestionInfo** によって与えられる問題のデータに基づいて最小の辞書を作る。冠詞類についてはどのような入力を与えられるかあらかじめ知りようもないからあらゆる変化形を登録することにするが、名詞については問題で与えられる名詞のみの変化形を登録する。冠詞類については、たとえば "dieserer" という誤った形に加え、それを（それなりに）正しく変形した "diesrer" なども捕捉できるようにした。実際のデータは以下のようなもの。

```

jenen =>
:article=>:origin=> "jener", :status=>["ms4", "mp3", "fp3", "np3",
"_p3"]

```

```

diesrer =>
:article=>
:origin=>"dieser",
:status=>
["ms1:article1",
"fs2:article1",
"fs3:article1",
"mp2:article1",
"fp2:article1",
"np2:article1",
"_p2:article1"]

```

上の例の "Mann" という名詞は、"Mann"、"Manns"、"Mannes"、"Männer"、"Männers"、"Männeres"、"Männern"、"Männeren"、"Mannsn"、"Mannsen"、"Mannesn"、"Mannesen"、"Mannn"、"Mannen" と、誤りを含む14の形で登録する。これ以外の形はタイプミスとして拒否される。実際のデータは以下の通りである。

```

:meaning=>"man",
"Mann"=>
  ["ms1",
   "ms2:noun1",
   "ms3",
   "ms4",
   "mp1:noun14",
   "mp2:noun14",
   "mp4:noun14",
   "mp3:noun14"],
"Manns"=>["ms2", "mp1:noun15", "mp2:noun15", "mp4:noun15",
"mp3:noun15"],
"Mannes"=>["ms2", "mp1:noun15", "mp2:noun15", "mp4:noun15",
"mp3:noun15"],
"Mannn"=>
  ["ms2:noun10",
   "ms3:noun10",
   "ms4:noun10",
   "mp1:noun10",
   "mp2:noun10",
   "mp4:noun10",
   "mp3:noun10",
   "mp3:noun16"],
"Männer"=>["mp1", "mp2", "mp4", "mp3:noun11"],
"Männers"=>["mp2:noun5"],
"Männeres"=>["mp2:noun5"],
"Männern"=>["mp3"],
"Männeren"=>["mp3:noun12"],
"Mannsn"=>["mp3:noun17"],
"Mannsen"=>["mp3:noun17"],
"Mannesn"=>["mp3:noun17"],
"Mannesen"=>["mp3:noun17"],
"Mannen"=>["mp3:noun16"]

```

入力された解答は **Answer** クラスのインスタンスとして、問題の情報とともに **Parser** クラスで処理される。

Parser においては、まず **Analyzer** クラスが、解答を、問題の情報とは無関係に分析する。たとえば "der Mann" という入力に対して、男性・単数・1格という当たり前の分析だけでなく、男性・複数・2格という分析も行わなければならない。学習者が名詞を複数形にするのを忘れている可能性があるからである。

次に **Parser** 中の **Comparer** クラスが、分析された情報を問題の情報と比較する。”der Mann” という入力に対し、`info:Mann:der:s:1` であれば「誤りなし」、`info:Mann:p:2` であれば「noun14」と判断することになる。

最後に **MessageMaker** が適切なメッセージを返す。

3. 結果

以上を実装するのに要したのは、約2000行であった。

実際の解析例をいくつか挙げておく。

No.1

`info = Mann:noun:der:s:1`

`ans = der Mann`

`fallacy=>0=>[], 1=>[]`

No.2

`info = Mann:noun:der:s:1`

`ans = des Mann`

`fallacy=>0=>[], 1=>["noun1", "np2"]`

No.3

`info = Mann:noun:der:s:1`

`ans = des Manns`

`fallacy=>0=>[], 1=>["np2"]`

No.4

`info = Mann:noun:der:s:2`

`ans = des Manns`

`fallacy=>0=>[], 1=>[]`

No.5

`info = Student:noun:mein:s:1`

`ans = meiner Student`

`fallacy=>0=>["article5"], 1=>[]`

No.6

`info = Student:noun:mein:p:1`

`ans = meiner Student`

`fallacy=>0=>[], 1=>["np4"]`

No.7**info = Student:noun:mein:s:2****ans = meines Students****fallacy=>0=>[], 1=>["noun7"]****No.8****info = Student:noun:mein:s:2****ans = meines Studentn****fallacy=>0=>[], 1=>["noun10"]****No.9****info = Frau:noun:mein:s:2****ans = unsrer Frau****fallacy=>0=>[], 1=>["np5"]****No.10****info = Frau:noun:unser:s:2****ans = unses Fraus****fallacy=>0=>["article4"], 1=>["noun6"]****No.11****info = Frau:noun:unser:s:2****ans = unsres Fraus****fallacy=>0=>[], 1=>["noun6"]****No.12****info = Leute:noun:dieser:p:3****ans = einer Leute****fallacy=>0=>[], 1=>["noun21", "np5"]**

返される結果はおおむね期待した通りのものとなっている。No.5とNo.6にあるように、解答の入力が同一であっても発せられるメッセージは異なることに注意してもらいたい。

唯一の問題は、No.6である。上で、np4の誤りが期待されるための条件として、「単複同形」を挙げていたが、「Student」はそうではない。しかし、np4は、格か数のどちらかが間違っていることを示すものだから、No.6にこの誤りを適用することは不適切とは言えない。「meiner Student」を単数・1格と解釈してarticle5を適用するか、複数・2格としてnoun14を適用するか、この場合にも決定することはできないからである（問題がたとえばp:4などである時には「格が違う」と正しく指摘することができる）。いずれにしてもこのような誤りが発生するのはきわめてまれであり（問題、解答をほぼランダムに組み合わせてテストしたところ、3000回に4回であった）、このような場合どのような（より曖昧でない）メッセージを返すのがよいかは、システムを運用した時の実際の出

現頻度を確認した上であらためて考えることにしたい。