動的な教材作成を目指して（3）：ドイツ語の動詞句を解析する

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/10.15017/19181

出版情報：言語文化論究. 26, pp.49-57, 2011-02-07. 九州大学大学院言語文化研究院
バージョン：
権利関係：

動的な教材作成を目指して（3）

── ドイツ語の動詞句を解析する ──

栗　山　　暢

1．目標と考え方

本稿は、初学者の書く誤ったドイツ語の文に、その誤りを指摘して改善させることを可能とするようなメッセージを返すことを目指すコンピュータプログラムの一部を解説するものである。

本稿で扱う動詞句とは、名詞などを含み動詞をその中心とする通常の意味での動詞句ではなく、動詞（本稿では動詞と言う時にはほとんどの場合助動詞を含む）と解釈されうる単語のみからなる一連の単語の連続のことである。文を解析するわけではないので、定形動詞の位置の確認は本稿の対象としないし、定形動詞の語尾についても、主語との一致は対象外とし、ただ、その時制のみを問題にする。分離動詞の分離した前つづりの位置も、文を解析する時にはじめて意味を持つことなので、ここでは対象としない。語順の自由を認めた上で唯一の正解を持つような問題を与え、それに与えられる答えを間違いを含むものとして解釈し正解へのヒントを与えるという目標は、名詞句を解釈した場合と同じである。”lernen”、”hat...gelernt”、”hätte...lernen sollen” などが、本稿の扱う動詞句である（本稿での説明では、定形は句の先頭に、それ以外の要素は ”...” の後に表記するものとする）。

たとえば、”hätte...lernen sollen” を正解とするような問題に対し、１）”sollte...lernen”、２）”hat...lernen sollen”、３）”soll...gelernt haben” などという答えが与えられることを予想して、それに正しいコメントを返すことが目標である。あるいは、４）”soll...lernt”、５）”habt...sollen lernen” など、ドイツ語初学者の書きそうなドイツ語を見慣れていない者には解答者の意図を推しはかることの困難な答えが与えられることもあるかも知れない。それぞれに対して与えようとするコメントは、次のようなものである。１）「sollen を完了形にせよ」２）「haben を接続法 II 式にせよ」３）「lernen ではなく sollen を完了形にせよ」４）「法助動詞が支配するのは原形である」５）「haben の現在人称変化は不規則。完了の助動詞に支配される過去分詞は文末に置かれる」。

まず最初にやるべきことは、ここでも、単語の形にかかわることである。”gelernt” が正しい場合に ”gelernet” と与えられるなど、考えられる誤った形をあらかじめすべて登録しておかなければならない。登録していない単語は単に拒否することになる、すなわちタイプミス同様それ以降の分析を行わないことになるので、ここでは可能な限り多くの誤った形を登録するのが望ましい。誤った形を導くための規則を、付録 A として巻末に示しておいた。この規則を使うと、”bringen” という動詞は、正しい形と誤った形をあわせて、図１の通り、32の変化形が登録されることになる。中にはおよそありそうもない形も含まれているが、今述べた理由により、それでよい。

bringe	bringene	bringst	bringest	bringenst	bringenest	bringt
bringet	bringent	bringenet	bringen	bringenen	brachte	bringte
bringete	brachtest	bringtest	bringetest	brachten	bringten	bringeten
brachtet	bringtet	bringetet	brächte	brächtest	brächten	brächtet
bring	gebracht	gebringt	gebringet			

図１　bringen の変化形

動詞句全体について把握しなけらばならない誤りは、主に、上に少し述べた通り、時制や語順や支配関係にかかわるものである。その一覧を付録 B に示した。どのようにしてそれらの誤りを把握するのか、次の項でやや詳しく説明する。

２．実　装

使用した言語は Ruby（version 1.9.1）。ファイル構成は以下の通り。”*” を付したもの以外はひとつのファイルがひとつの（Ruby の）クラスに対応し、ボールド体のものが今回加わった部分である。./helper および ./object_retriever 以下はテスト用のコードであるから、本稿で扱うのは、VerbInfo、VerbPhrase、VpComparer の３つのクラスである。

./beckmesser.rb
./bm_require.rb*
./constants.rb*
./lib/extension.rb*
./lib/dictionary.rb
./lib/dictionary/article_data.rb*
./lib/dictionary/make_noun_info_from_line.rb*
./lib/dictionary/make_adj_info_from_line.rb*
./lib/dictionary/make_verb_info_from_line.rb*
./lib/question_info.rb
./lib/question_info/noun_info.rb
./lib/question_info/verb_info.rb
./lib/question_info/verb_info/aux_sequence.rb
./lib/answer.rb
./lib/answer/word.rb
./lib/answer/article.rb
./lib/answer/noun.rb
./lib/answer/adjective.rb
./lib/parser.rb
./lib/parser/structure.rb
./lib/parser/analyzer.rb
./lib/parser/noun_phrase.rb
./lib/parser/verb_phrase.rb
./lib/parser/comparer.rb
./lib/parser/np_comparer.rb
./lib/parser/vp_comparer.rb
./lib/message_maker.rb
./helper/object_retriever.rb*
./object_retriever/or_parser.rb*

./object_retriever/or_verb_phrase.rb*
./object_retriever/or_vp_comparer.rb*
./object_retriever/or_message_maker.rb*

２．１ VerbInfo

VerbInfo は、問題として与える動詞句の情報を処理するクラスであり。法、時制、使用する助動詞、態、（本稿では扱わないが）主語に関する情報を保持する。時制については、普通行われている「６時制」では助動詞の処理が困難なので、現在時制と過去時制のふたつのみを認めることにし、完了時制と未来時制はそれぞれ完了と未来の助動詞を使用する、と記述する。たとえば、

”lernen:verb:ind_past,ind_pres(perf):act:1s”

という記述で、”:” で区切った三番目の項が法と時制（この場合、直説法（ind）過去（past）と直説法現在（pres）のどちらか。ほかに法は、sub1= 接続法 I 式、sub2= 接続法 II 式、imp_s=du に対する命令法、imp_p=ihr）、次の項が態（act= 能動態、pas= 受動態、st_pas= 状態受動）を表すものとする。助動詞は法と時制の後の括弧の中に、内側（定形から遠い方）から並べて示す（法助動詞は原形、未来は ”future”、完了は ”perf”）。ind_pres（müssen.future’）であれば、”wird...lernen müssen” ないしは ”muss...lernen” を要求していることを表現している（future の右肩の ”’” はその助動詞があってもなくてもよいことを示している）。このデータにもとづいてプログラムが実際に生成するデータは、次のようなものである（一部を略した）。

```
#<Beckmesser::QuestionInfo::VerbInfo:0x00000000bbc1f0
@aux_sequence=
  {”ind_past”=>{},
   ”ind_pres”=>
    {0=>{:can_be_omitted=>false, :kind=>:perfect, :master=>:finite}}},
@has_aux={”ind_past”=>false, ”ind_pres”=>true},
@inf_form=”lernen”,
@mood=[”ind”],
@mood_tense=[”ind_past”, ”ind_pres”],
@origin=”lernen”,
@passive_aux=nil,
@person_number=[”1s”],
@src=”lernen:ind_past,ind_pres(perf):act:1s”,
@tense=[”past”, ”pres”],
@voice=:active>
```

２．２ VerbPhrase

VerbPhrase クラスでは、ひとつの文（ひとつの主文あるいは副文）の動詞句を、できるだけ問題の情報に頼らずに解析することを目指す（基本となる動詞や主文か副文かという情報はやむを得ず使わざるを得なかった）。

まずしなければならないのは、基本となる動詞（助動詞として働いていない動詞）と定形動詞を決定することである。答えの中に出現する動詞がひとつだけの場合は両者がともにその単語であろ

うことは自明であるが（それでも使われるべき動詞が使われていないなどという誤りのある可能性はある）、動詞がふたつ以上現れる場合にはなかなか複雑なことになる。動詞がふたつの場合は比較的簡単そうに見えるけれども、たとえば ”hat...gehabt” のように、基本となる動詞と助動詞が同じ単語である場合などは、それほど単純に決定できるものではない（”haben...haben” などは、どちらを定形と考えればよいのだろうか）。

与えられる答えには誤りが含まれていることを前提としているのだから、語順、語形のいずれかに決定的な役割を負わせることはできない。極端な例を挙げれば、”er wird krank sein” という答えの目指すものが実は ”er ist krank geworden” であるようなことを常に考慮に入れておかなければならない。一番複雑な場合、つまり、基本となる動詞が文中に複数回出現する可能性のある ”haben”、”sein”、”werden” であり、動詞の数が３個以上、しかも基本となる動詞が複数回出現している場合についてのみ、簡単に説明する。

基本となる動詞は、次の規則を順に適用して、最初に条件が満たされたところで決定する。

1）当該の動詞のうち、過去分詞としか解釈できない形をしているものがひとつだけであればそれを基本の動詞とする（”hat...gehabt” においても ”gehabt...hat” においても ”gehabt” を動詞、”hat” を助動詞と解釈する）。
2）当該の動詞のうち、定形ではあり得ない形をしているものがひとつだけであればそれを基本の動詞とする（”ist...sein” であれば、”sein” が動詞、”ist” が助動詞）。
3）主文であることが期待される時には（問題の情報にそう記述があれば）動詞句の中で２番目以降最初に現れるものを基本の動詞とする（”wird...sein sein”、”sein...sein werden” の両者とも最後から２番目の ”sein”）。
4）副文であることが期待される時には、基本の動詞が ”sein” の場合には最初の ”sein”（”sein sein werden” において最初の ”sein”）、”haben” ないし ”werden” の時は、動詞句の最後が ”haben”（”werden”）の時は最初の（”haben sollen haben”）、そうでない時は（”haben haben sollen”）二番目以降最初のものを基本の動詞とする。

これをもとに、動詞句の語順を次の規則に従って決定する（基本となる動詞が先頭、定形動詞が末尾になるように並べる）。

1）基本となる動詞が動詞の列の先頭にある場合には、動詞の列をそのまま動詞句の語順とする（”lernen...sollen haben”=>”lernen sollen haben”。最後の ”haben” が定形）。
2）基本となる動詞が動詞の列の末尾にある時には、動詞の列の逆順を動詞句の語順とする（”haben...sollen lernen”=>”lernen sollen haben”）。
3）基本となる動詞が動詞句の２番目にある時には、２番目から末尾までを順に並べ最後に１番目の単語を置いたものを動詞句の語順とする（”haben...lernen sollen”=>”lernen sollen haben”）。
4）それ以外の場合は基本となる動詞から末尾までを順に並べその後に先頭から基本となる動詞の直前までを逆順に並べる（”werden...haben lernen sollen”=>”lernen sollen haben werden”）。

１）は（正しく）定形が後置された場合、３）は（正しく）定形が２番目におかれた状態を想定し、２）と４）は語順に関する規則が守られていない場合を想定している（その意味で２）と４）は便宜的に語順を決定しているにすぎない）。

これで基本となる動詞から始まって定形動詞にいたる動詞句の語順が決定された。今度はそれを逆にたどって、単語の形と助動詞の接続のしかた（完了の ”haben” が正しく過去分詞を支配しているか、など）をチェックする。単語の形に関しては、接続関係が正しくない場合にはその単語の品詞（原形、定形、分詞など）を決定できない場合があるので、品詞が決定できる時、あるいは品詞は決定できなくても誤りの種類がひとつである時にのみチェックする。問題の情報を ”info” とし、答えを ”ans” とすると、

info = ”Mann:noun:der:s:1::Frau:noun:der:s:4::lernen:verb:ind_past:act:1s”

ans = ”der Mann wird die Frau gelernet”

の場合、次のようなデータが作られることになる（文脈上不必要な部分は省略した）。

```
{:finite_pos=>2,
 :base_pos=>5,
 :sequence=>[5, 2],
 :finite_status=>[”ind_pres_3_s”],
 :finite_word_fallacy=>{”ind_pres_3_s”=>[]},
 :voice=>[:active, :passive],
 :connection_fallacy=>
   {:active=>{2=>[], 5=>”vp2(future)”}, :passive=>{2=>[], 5=>[]}},
 :word_fallacy=>
   {:active=>{2=>[], 5=>[”verb16”]}, :passive=>{2=>[], 5=>[”verb16”]}},
 :aux_kind=>{:active=>{2=>:future}, :passive=>{2=>:passive}}},
```

すぐ後で見るようにこの後の解析の対象とするのは問題の情報によって与えられた態だけであるが、受動態で書くべきところを能動態で書いたような誤りを今後把握しようとすることがないとは言えないので、受動態である可能性がある時には問題の情報がなんであれ解析することにしてある。

基本となる動詞、定形動詞の位置は、それぞれ :base_pos、:finite_pos で示されいる（先頭は 0 である）。:sequence がその並び順。:connection_fallacy が助動詞の支配関係に関する誤り、:word_fallacy が単語の形に関する誤りである。答えの中の ”gelernet” は ”gelernt” としか解釈できないので（”verb16” がそれを示している）、答えは ”wird...gelernt” という組み合わせだということになり、受動態だと接続が正しいこと、能動態だと接続が誤っていることが把握されている（”vp2(future)”。”future” はこの単語が未来の助動詞に支配されるべきであることを示している）。

2.3 VpComparer

本稿では触れていないが、与えられた答えの中に複数の品詞であり得る単語がある場合には（たとえば ”sein” は動詞と不定冠詞類のふたつの品詞であり得る）、文の構造も複数存在することになる。それに応じて VerbPhrase（のインスタンス）も複数生成することになるが、VpComparer では、まず生成した複数の VerbPhrase の内容を（問題の情報を適宜参照しながら）解析しもっとも適切だと思われるものを選び（すなわち複数の構造からもっとも適切だと思われるものを決定し）、その構造について詳細に問題の情報と照合する。本稿では後者についてのみ説明する。

まず、助動詞の並び方を調べる。これは単に、問題の情報から可能な助動詞の並び方をすべて生

成し、そのひとつと VerbPhrase に保持されている当該の情報（上の :aux_kind）とが一致するか調べればよい。:aux_kind では法助動詞は区別せずにただ :modal としてあるだけだから、一致するものが見つかった場合、個々の法助動詞が問題によって要求されているものかどうかさらにチェックする。

一致するものが見つからないということは、答えの文の構造が問題によって要求されているものと比較的大きく異なっているということである。この場合、使用すべき助動詞を使っていなかったり使用すべきでない助動詞を使っていたりしないかどうかまずチェックして、そうであればそれを誤りとして登録し、そうでなければ順番に助動詞の有無をチェックして、最初に見つかった誤りを登録する。一度にすべての誤りを検知できればそれに越したことはないように思われるけれども、この段階になっては人間の目で判断しても解答者の意図がわからないことが多いだろうということは容易に予想されるし、実際上は助動詞の数が 2 を上回ることもほとんどないであろうから、この方法で十分に機能することが期待される。

最後に定形の法と時制をチェックして、VerbInfo、VerbPhrase とともに名詞句の解析の時にも使用した MessageMaker に渡す。上と同じ問題、答えの組み合わせで得られるデータは次のとおり（直接関係ないところは省略した）。

```
{:extra_vp=>false,
 :finite=>
  {:mood_tense_right=>false,
    :mood_right=>true,
    :tense_right=>false,
    :mood_and_tense_ambiguous=>false,
    :word_fallacy=>[]},
:aux_fallacy=>
  {:position=>{5=>[], 2=>["vp29"]},
    :unused_modal_aux=>[],
    :extra_modal_aux=>[]},
:aux_sequence_right=>false}
```

:mood_right=>true が法が問題と合致していること、:tense_right=>false が時制が合致していないことを示し、:aux_fallacy の "vp29" が使われるべきでない助動詞が使われていることを示している。

MessageMaker では、単語の形に関する誤りがある場合にはまずそれだけを指摘して、単語の形がすべて正しい時にのみほかの誤りを指摘する。"er wird...lernen" が正解である時に "er wird...gelernet" という答えが与えられた場合、"werden" が未来の助動詞であり原形を支配することを指摘するよりも前に、おそらく過去分詞のつもりであるだろう "gelernet" の形の間違いを指摘すべきであろうと考えるからである。

3．結　果

VerbInfo、VerbPhrase、VpComparer を実装するのに約1200行を要した。

実際の解析例をいくつか紹介する。message の中、"=>" の前の数字は答えの中の単語の位置（0

から始まる）を示している。答えを文の形で与えてあるが、使われている名詞やまして文の意味はまったく意味を持たず、ただ動詞句にのみ注目していただきたい。

---

info = ”Mann:noun:der:s:1::Frau:noun:der:s:4::lernen:verb:ind_past:act:1s”
(”der Mann lernte die Frau”)
no.1 ans = ”der Mann lernte die Frau”
message = {2=>[]}
no.2 ans = ”der Mann lernt die Frau”
message = {0=>[], 1=>[], 2=>[”vp21”], 3=>[], 4=>[]}
no.3 ans = ”der Mann hat die Frau gelernt”
message = {5=>[], 2=>[”vp29(perfect)”]}
no.4 ans = ”der Mann wird die Frau lernen”
message = {5=>[], 2=>[”vp29(future)”]}
no.5 ans = ”der Mann habt die Frau gelernet”
message = {2=>[], 5=>[”verb16”]}

---

no.1は正解、no.2は時制が間違っている、no.3、no.4はそれぞれ完了形、未来形にしてはいけない、というメッセージ。no.5は ”gelernet” という誤った形をとらえている。

---

info = ”Mann:noun:der:s:1::Frau:noun:der:s:4::lernen:verb:ind_pres(perf.wollen’):pas”
no.6 ans = ”der Mann ist die Frau gelernt worden”
message = {5=>[], 6=>[], 2=>[]}
no.7 ans = ”der Mann will die Frau gelernt worden sein”
message = {5=>[], 6=>[], 7=>[], 2=>[]}
no.8 ans = ”der Mann ist die Frau gelernt geworden”
message = {2=>[], 6=>[”verb50”], 5=>[]}
no.9 ans = ”der Mann hat die Frau gelernt geworden”
message = {2=>[], 6=>[”vp9”], 5=>[]}
no.10 ans = ”der Mann wird die Frau gelernt”
message = {5=>[], 2=>[”vp27(perfect)”]}
no.11 ans = ”der Mann wird die Frau gelernt wollen”
message = {5=>[”vp27(passive)”], 6=>[], 2=>[]}
no.12 ans = ”der Mann soll die Frau gelernt werden”]
message = {5=>[], 6=>[], 2=>[”vp29(modal)”]}

info = ”Mann:noun:der:s:1::Frau:noun:der:s:4::lernen:verb:ind_pres(wollen.perf):act”
(der Mann hat die Frau lernen wollen)
no.13 ans = ”der Mann will die Frau gelernt haben”
message = {:fallacy=>{5=>[”vp27(modal)”], 6=>[], 2=>[]}}

---

no.6と no.7が正解（”wollen” があってもなくてもいいため）。no.8は ”werden” が受動の助動詞とした使われる時の過去分詞の誤りを、no.9は ”werden” の完了の助動詞の誤りをとらえている（no.9では２番目の ”hat” ではなく ”werden” の方について誤りを指摘している。助動詞 ”haben” の問

題ではなく、”werden” の問題だからである)。no.10は ”werden” を完了形にすべきことを、no.11は ”gelernt” が受動の助動詞に支配されるべきであることを指摘している。no.12は使われるべきでない助動詞 ”sollen” が使われていることを示している。

いずれも完全に期待通りの結果が得られた。現状で把握できている唯一の問題は、no.5に使わている ”habt” について、誤りの捕捉がなされていないことである。これはこの形が法に関して複数の可能性を持つためであるので今のところでは仕方がない。no.5の答えであればこれが ”hat” でなければならない（すなわち verb36を指摘しなければならない）ことはほぼ自明であるから、次の段階で文の解析をする時に主語との語尾の一致をチェックした上で正しく対処できるようにしたい。

附録 A　動詞の形に関する誤り

```
VERB_WANDERETE_ERROR = ”verb1”
VERB_WARTTE_ERROR = ”verb2”
VERB_REGNTE_ERROR = ”verb3”
VERB_GEAUFMACHT_ERROR = ”verb4”
VERB_GEBESUNGEN_ERROR = ”verb5”
VERB_GEGANGEN_WITHOUT_PREFIX_ERROR = ”verb5.5”
VERB_AUFMACHT_ERROR = ”verb6”
VERB_BEGESUNGEN_ERROR = ”verb7”
VERB_GEANVERTRAUT_ERROR = ”verb8”
VERB_ANGEVERTRAUT_ERROR = ”verb9”
VERB_LERNETE_ERROR = ”verb10”
VERB_MUST_NOT_BE_WEAK_ERROR = ”verb11”
VERB_ANVERGETRAUT_ERROR = ”verb12”
VERB_GEWARTT_ERROR = ”verb13”
VERB_GEREGNT_ERROR = ”verb14”
VERB_GESTUDIERT_ERROR = ”verb15”
VERB_GELERNET_ERROR = ”verb16”
VERB_IMP_S_WANDER_ERROR = ”verb17”
VERB_IMP_S_GEBE_ERROR = ”verb18”
VERB_IMP_P_WARTT_ERROR = ”verb19”
VERB_IMP_P_TUET_ERROR = ”verb20”
VERB_IMP_P_SEIT_ERROR = ”verb21”
VERB_IMP_P_LERNET_ERROR = ”verb22”
VERB_IMP_P_SOME_ERROR = ”verb23”
VERB_SUB2_NO_UMLAUT_ERROR = ”verb24”
VERB_SUB2_WRONG_UMLAUT_ERROR = ”verb25”
VERB_SUB2_SCHRIE_ERROR = ”verb26”
VERB_SUB2_NO_E_TAIL_ERROR = ”verb27”
VERB_RIGHT_SUB2_EXIST_ERROR = ”verb28”
VERB_SUB2_SOME_ERROR = ”verb29”
VERB_PRESENT_WANDER_ERROR = ”verb30”
VERB_PRESENT_HANDELE_ERROR = ”verb31”
VERB_PRESENT_REISEST_ERROR = ”verb32”
VERB_PRESENT_SEIN_NOT_WEAK_ERROR = ”verb33”
VERB_PRESENT_TUEN_ERROR = ”verb34”
VERB_PRESENT_EXTRA_E_ERROR = ”verb35”
VERB_PRESENT_HABEN_2S_3S_ERROR = ”verb36”
VERB_PRESENT_NO_EXTRA_E_ERROR = ”verb37”
VERB_PRESENT_GEBST_ERROR = ”verb38”
VERB_PRESENT_IRREGULAR_ERROR = ”verb39”
VERB_PRESENT_HAELTET_ERROR = ”verb40”
```

```
VERB_PRESENT_LERNENST_ERROR = ”verb41”
VERB_SUB1_SEIE_ERROR = ”verb42”
VERB_SUB1_3_S_SEIT_ERROR = ”verb43”
VERB_SUB1_2_P_SEIT_ERROR = ”verb44”
VERB_SUB1_1_3_P_SEIN_ERROR = ”verb45”
VERB_SUB1_HANDELE_ERROR = ”verb46”
VERB_SUB1_HANDELEST_ERROR = ”verb47”
VERB_SUB1_HANDELEN_ERROR = ”verb48”
VERB_SUB1_2_P_HANDELET_ERROR = ”verb49”
VERB_GEWORDEN_IN_PASSIVE_PERFECT_ERROR = ”verb50”
```

附録 B　動詞句に関する誤り（一部本稿とは無関係なものもまじっている）

```
VP_NO_VP_ERROR = ”vp0”
VP_SEQUENCE_AMBIGUOUS_ERROR = ”vp1”
VP_CONNECTION_ERROR = ”vp2”
VP_NO_FINITE_ERROR = ”vp3”
VP_MULTIPLE_BASE_VERB_ERROR = ”vp4”
VP_NO_BASE_VERB_ERROR = ”vp5”
VP_PAST_P_AS_FINITE_ERROR = ”vp6”
VP_MUST_BE_INFINITIVE_ERROR = ”vp7”
VP_MUST_BE_PAST_P_ERROR = ”vp8”
VP_PERFECT_AUX_MUST_BE_SEIN_ERROR = ”vp9”
VP_PERFECT_AUX_MUST_BE_HABEN_ERROR = ”vp10”
VP_STATAL_PASSIVE_PERFECT_ERROR = ”vp11”
VP_MUST_BE_STATAL_PASSIVE_ERROR = ”vp12”
VP_MUST_NOT_BE_STATAL_PASSIVE_ERROR = ”vp13”
VP_PERFECT_FUTURE_ERROR = ”vp14”
VP_MULTIPLE_PASSIVE_AUX_ERROR = ”vp15”
VP_MULTIPLE_FUTURE_AUX_ERROR = ”vp16”
VP_MULTIPLE_PERFECT_AUX_ERROR = ”vp17”
VP_FINITE_POSITION_TO_TOP_ERROR = ”vp18”
VP_FINITE_POSITION_TO_SECOND_ERROR = ”vp19”
VP_FINITE_POSITION_TO_LAST_ERROR = ”vp20”
VP_TENSE_ERROR = ”vp21”
VP_MOOD_ERROR = ”vp22”
VP_MOOD_TENSE_AMBIGUOUS_ERROR = ”vp23”
VP_WRONG_AUX_ERROR = ”vp24”
VP_AUX_PERFECT_OUTER_TO_INNER_ERROR = ”Vp25”
VP_AUX_PERFECT_INNER_TO_OUTER_ERROR = ”vp26”
VP_AUX_MASTER_ERROR = ”vp27”
VP_NO_AUX_ERROR = ”vp28”
VP_EXTRA_AUX_ERROR = ”vp29”
```

参考文献

1．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『言語文化論究』No.25 p.45-52.（2010. 3. 26）

2．動的な教材開発を目指して（2）── ドイツ語の名詞句を解析する『ドイツ語情報処理研究』（ドイツ語情報処理学会）No.20 p.25-35.（2010. 3. 31）