The Generation of Idiomatic and Collocational Expressions

Johannes Matiasek
Austrian Research Institute for Artificial Intelligence*
Schottengasse 3,
A-1010 Vienna, Austria
Email: john@ai.univie.ac.at

Abstract

Collocations whose semantic content is not or only partially composed from the semantic content of their parts are often viewed as problematic for generation. In this paper a tactical generator combining FUF as the generation engine and HPSG as the grammar framework is presented. It is shown, that the lexicon driven approach to syntactic and semantic processing is well-suited for the generation of idioms exhibiting various degrees of noncompositionality and syntactic restrictions.

1 Introduction

Co-occurrences of words that cannot be characterized by structural (*syntactic*) rules alone but depend to a large extent on the presence of specific *lexical* items are commonly referred to as collocations. Their dependence on particular words qualifies them as a lexical phenomenon, the difficulties to reveal the semantic content of a collocation from the semantics of its parts has established them as a challenge for compositional semantics. Although there have been attempts to analyze idioms compositionally and it has been proven, that any semantics can be reformalized as a compositional one [Zadrozny, 1992], the traditional view of idioms as entities whose meaning cannot be derived from the meaning of its parts is still advocated.

From a practical point of view—when considering the requirements a tactical generation component has to fulfill—this debate is not especially relevant. The tactical generator that will be described here is part of a multilingual text generation system and thus is constrained by the interface definitions of the overall architecture of the system. Since output in different languages has to be produced, the strategic planner cannot be explicitly geared towards the particularities of the output languages w.r.t. idiomatic and collocational expressions. Instead, the input for the tactical generators is largely a *semantic* specification in a SPL-based language (cf. [Kasper, 1989]) and specifies the functional structure of the text to be generated at a level not fine-grained enough to adopt the compositional view mentioned above.

Precisely for this noncompositionality and the dependence on particular lexical items, the generation of collocations is regarded as a problem in the generation literature (cf. [Matthiessen, 1991]). Especially generation frameworks (as, e.g., systemic functional linguistics) that derive the syntactic and semantic behavior of words from their place in some hierarchy have difficulties in accounting for the exceptional characteristics of these words in the context of idiomatic usage.

The rest of this paper is organized as follows: first, an overview of the collocation phenomena as they occur in German will be given, then the basic architecture of the tactical generator will be reviewed. The main part of the paper is devoted to show, how the chosen formalism and its implementation is capable to account for the various collocation phenomena encountered in German.

2 Idioms and other Collocations

The degree of compositionality varies among different collocational expressions. The meaning of "unanalyzable" idioms such as

(1) den Löffel abgeben
"kick the bucket"

is totally unrelated to the meaning of its parts. In metaphorical idioms such as

(2) eine Behauptung angreifen
"to attack a claim"

at least part of the idiom is accessible referentially for modification or quantification. A third class of collocations, the so-called *support verb constructions* combine a relational noun with a "light verb" which contributes only little to the meaning of the phrase, whereas the core meaning is provided by the noun, as, e.g.

*The work reported here has been carried out within the LRE Project *GIST* (LRE 062-09) and funded by the Austrian *Forschungsförderungsfonds der Gewerblichen Wirtschaft*, Grant 2/329. Financial support for the Austrian Research Institute for Artificial Intelligence is provided by the Austrian *Bundesministerium für Wissenschaft, Forschung und Kunst.*

(3) einen Antrag (auf etwas) stellen[1]
"to apply (for something)"

In [Krenn and Erbach, 1994] a more complete classification of idioms and support verb constructions is presented and a treatment within the HPSG framework is given. For our present purpose it is sufficient to show, which phenomena need to be handled.

Although a fixed wording is characteristic for idioms and support verb constructions, some variations may occur. The degree of variability, however, depends on the particular collocation. Most notably, almost all idioms allow for inserting material (e.g. negation particles) between the words constituting the idiom or topicalization of a "frozen" complement (see (4)), so simply storing idioms as multi-word strings in the lexicon is not an option.

(4) a. den Löffel nicht abgeben
"not to kick the bucket"

b. Den Löffel hat er abgegeben
"he has kicked the bucket."

Other syntactic operations may or may not be possible. Passivization often causes the idiomatic reading to be lost, as, e.g., in

(5) # der Löffel wird abgegeben[2]
"the spoon is handed over"

The same may hold for modification. E.g.,

(6) # Peter gab den silbernen Löffel ab.
"Peter handed over the silver spoon."

can only be interpreted literally, whereas in

(7) Peter gab den Löffel vorzeitig ab.
"Peter kicked the bucket prematurely."

the idiomatic reading is retained. Other idioms, such as, e.g., (2) allow modification of the "frozen" complement and also retain the idiomatic reading when passivized.

Thus we have the following requirements on the tactical generator (and the grammar) in order to be able to deal with idioms and support verb constructions correctly:

- It must be able to represent "frozen" complements in the lexicon.
- It must be possible to assign particular lexemes to frozen complements.
- Frozen complements must be represented in such a way, that the usual syntactic machinery such as case assignment, topicalization or passivization can operate on them.
- Nevertheless means must be provided to inhibit certain operations (e.g. modification or passivization) in case of idioms that lose their idiomatic reading on such operations.

[1] as far as possible the examples here are taken from the domain of administrative texts, the application domain of the GIST system.

[2] # is used to indicate that only the literal reading is available.

3 HPSG in FUF

The tactical generator described here has been developed in the context of a multilingual text generation system. One of the objectives of the project is to reuse existing resources if such existing resources are appropriate. For the German tactical generator an implementation of an HPSG[3] style grammar of German (used for parsing and generation, but on a different software platform) and a morphology module were available inhouse [Trost, 1991; Trost and Matiasek, 1994]. A LISP-based generator was available as public domain software, namely the FUF package [Elhadad, 1991]. The integration task of these components and the working of the generator is described more throughly in [Matiasek and Buchberger, 1995], here only the aspects relevant for the generation of idiomatic expressions—especially the way lexically driven processing is implemented within FUF—will be sketched.

3.1 The FUF generator

FUF [Elhadad, 1991] implements a surface generator for natural language. It is based on the theory of functional unification grammar [Kay, 1979] and employs both phrase structure rules (encoded by a special CATegory feature) and unification of feature descriptions. Input to FUF is a partial feature description constraining the utterance to be generated. The output of FUF is a fully specified feature description (in the sense of the particular grammar) subsumed by the input structure, which is then linearized to yield a sentence.

Grammar and lexicon are specified as one large feature description, containing at least one disjunction (given by the `alt` keyword) ranging over the phrasal and lexical categories of the grammar. The feature `cat` is used to indicate these categories. Strings are associated with lexical categories via the feature `lex`. Pointers can be used to enforce identity of substructures and provide a means to percolate information within a feature structure.

Generation with FUF starts from an input feature structure constraining the utterance to be produced. FUF unifies the grammar *into* the input structure, i.e. enriches and further constrains it. Alternatives are explored sequentially until one branch succeeds.

When unification at the current level is complete, i.e. nothing further can be added to the input structure, recursion on the subconstituents is performed. The constituents can be given implicitly or via the special feature `cset`. Every substructure of the enriched input structure which represents a category is recursively unified with the grammar. This process is repeated in a breadth first fashion until all constituents are leaves, i.e. have no subconstituents.

To determine which substructures correspond to constituents needing recursion, FUF employs two methods. The default strategy collects all substructures of the current level bearing the category feature

[3] Head Driven Phrase Structure Grammar [Pollard and Sag, 1987; 1994]

$$\left[\begin{array}{ll} \text{PHON} & \textit{"geht"} \\ \text{SYNSEM} & \left[\begin{array}{ll} \text{LOC} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & {}_{verb}\left[\begin{array}{ll} \text{VFORM} & \textit{finit} \\ \text{TENSE} & \textit{present} \\ \text{PERSON} & \textit{third} \\ \text{NUM} & \textit{sg} \end{array}\right] \\ \text{SUBCAT} & \left\langle {}_{synsem}\left[\begin{array}{ll} \text{LOC} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & {}_{noun}\left[\begin{array}{ll} \text{CASE} & \textit{nom} \\ \text{PERSON} & \textit{third} \\ \text{NUM} & \textit{sg} \end{array}\right] \\ \text{SUBCAT} & \langle\rangle \end{array}\right] \\ \text{CONT} & {}_{nom\_obj}\left[\text{INDEX}\ \boxed{1}\right] \end{array}\right] \end{array}\right] \right\rangle \end{array}\right] \\ \text{CONT} & {}_{psoa}\left[\begin{array}{ll} \text{REL} & \textit{walk} \\ \text{ACTOR} & \boxed{1} \end{array}\right] \end{array}\right] \\ \text{NONLOC} & \textit{nonloc} \end{array}\right]_{synsem} \end{array}\right]_{word}$$

Figure 1: Lexical Entry for *"geht"* in HPSG

`cat`. The second strategy requires explicit specification of the subconstituents in the grammar by means of the special feature `cset` (constituent set). If such a feature is present, FUF does not resort to its default strategy but only performs recursion on these explicitly given substructures.

FUF offers the possibility to define an index on an alternative specification using a `:index <path>` declaration. To avoid blind search in a situation where the index feature is not yet specified, the decision which branch to take can be delayed using the `:wait <path>` declaration. Then unification proceeds ignoring the disjunction at hand and puts the delayed choice point onto an agenda. Processing of the delayed disjunctions continues if either the value waited for is present or if nothing else is left to do.

The recursive unification process handles only the dominance relations of the grammar. In order to account for linear ordering of the resulting tree shaped feature structure, FUF performs a linearization process after unification has finished. Linear precedence of constituents must be specified in the grammar using the special feature `pattern`. Only constituents mentioned in a pattern are realized during linearization. Linearization traverses the tree, extracts the strings found in the `lex` feature of the leaves, and flattens this structure according to the `pattern` directives found.

3.2 HPSG

In HPSG [Pollard and Sag, 1987; 1994] the fundamental objects of linguistic analysis are signs modeled by typed feature structures and constrained by global principles. The basic attributes for signs include PHON for phonological information and SYNSEM for syntactic and semantic information. SYNSEM in turn is highly structured including LOCal and NONLOCal features. LOCal features comprise CONTent, containing semantic information and the CATegory complex, which includes the HEAD features and the SUBCAT list to model subcategorization information. NONLOC features are used to model nonlocal dependency constructions such as topicalization, questions and relative clauses. As an example of how the HPSG encoding of syntactic and semantic information in the lexicon looks like, the lexical entry of the German verb *geht* is given in Fig. 1 above.

HPSG does not employ phrase structure rules. Instead, very general dominance schemata are given. Which arguments a lexical head takes is lexically specified in its SUBCAT list. Also adjunction is specified lexically; the adjunct is seen as the semantic head which selects the kind of signs it modifies, the modified sign remains the syntactic head of the resulting phrase. Long distance dependencies are handled in HPSG not in terms of movement but via structure sharing of the values of a SLASH feature percolating the "moving" constituent.

3.3 Lexicon Driven Processing in FUF

The main obstacle of directly implementing HPSG in FUF is the category driven top-down processing scheme employed by FUF, whereas HPSG encodes phrase structure mainly in the lexicon and thus lends itself better to a bottom-up generation strategy. Since the control regime of FUF cannot be changed in principle (only delay methods are available), the grammar itself has to account for adequate processing characteristics. Thus the lexicon driven approach has to be emulated within the grammar, taking the operational behavior of FUF into account.

The basic idea for realizing head driven processing behavior is to use the `cset` and `pattern` special attributes of FUF in an asymmetrical fashion. Generation of a phrase starts by realizing its `head-dtr`. Therefore only the head daughter is specified in the constituent set of the phrase, i.e. `(cset (head-dtr))`. Once the lexical head of the phrase is generated, its argument list is activated using the default recursion strategy of FUF (since no `cset` attribute is present). The lexically specified arguments are now generated in a (virtually) bottom up fashion. Structure sharing percolates the `args` upwards to the phrasal level, where they are then realized via the pattern feature. The basic mechanism of encoding this

```
((cat      phrase)
 (head-dtr ((cat lex-cat) ... ))
 ;; --- percolate arguments
 (args     {^ head-dtr args})
 ;; --- recurse only on head daughter
 (cset     (head-dtr))
 ;; --- realize head and arguments
 (pattern  (args head-dtr)))
```

Figure 2: Head driven generation in FUF

processing strategy in the grammar is given in Fig. 2. If functional categories are present in a phrase, then the appropriate slots have to be specified and added to `cset` and `pattern`.

Thus the `args` feature serves the same purpose as the SUBCAT list in standard HPSG, but instead of subcategorizing only for *synsem* values as proposed in [Pollard and Sag, 1994] the convention of [Pollard and Sag, 1987] is adopted and the whole sign is subcategorized for. Thus not only a simpler structure—corresponding more closely to the FUF assumptions on grammar layout—is obtained, but also the basic requirements for encoding idioms in the lexicon are met (cf. [Krenn and Erbach, 1994]).

The shape of the resulting phrase largely depends on the kind of arguments its lexical head admits. In order to realize its arguments, every word able to act as the head of a phrase has to provide a syntactic and semantic specification of its arguments in the lexicon. This specification also has to account for possible long distance phenomena, i.e. extraction of an argument (e.g., wh-movement). Furthermore, variations of case assignment (e.g., in passivization) have to be accounted for .

An example for the lexical specification of argument structure is given in Fig. 3 showing the actual encoding of the lexical entry for the verb *"beantragen"* ("claim"), subcategorizing for an actor and an actee[4]. By this lexical specification the valid sentences with head verb *beantragen* can be generated, using the generation scheme in Fig. 2

4 Lexically Driven Generation of Idioms in FUF

Given the machinery for lexically driven generation sketched above, we can now turn to the representation of collocational phrases in the lexicon. The representation of the support verb construction *"einen Antrag stellen"* is show in Fig. 4 below. This phrase is synonymous to *"beantragen"* (cf. Fig. 3) and thus the semantic representation as far as the input specifications are concerned should be the same. Thus, from the input

[4]`#(external np-ext-da)` is a macro specification in FUF, which the grammar expands to a nominative (subject) NP in case of an active sentence and an optional PP_{von} in case of passivization. `#(external np-int)` expands to NP_{acc} (active) or a subject NP_{nom} (passive). See [Heinz and Matiasek, 1994] for a theoretical background and [Matiasek and Buchberger, 1995] for implementation details

```
((cat     lex-verb)
 (lxm     "beantrag")
 (concept apply)
 (args
  ((actor   #(external np-ext-da))
   (actee   #(external pp-int)))))
```

Figure 3: Lexical Entry for *"beantragen"* in FUF

```
((cat     lex-verb)
 (lxm     "stellen")
 (concept apply)
 (args
  ((actor    #(external np-ext-da))
   ;; --- pointer to range of instance
   (actee    {^ instance args range})
   ;; --- "frozen" complement
   (instance #(external pp-int)
             ((lxm "Antrag")
              (concept application))))))
((cat     lex-noun)
 (lxm     "Antrag")
 (concept application)
 (args    ((range   #(external pp-auf)))))
```

Figure 4: Lexical Entries for *"Antrag stellen"* in FUF

```
(8) ((concept apply)
     (args
      ((actee
        ((concept retirement-pension))))))
```

both *"Alterspension beantragen"* and *"Antrag auf Alterspension stellen"* can now be generated. However, the `instance` slot in Fig. 4 accounts for the differences between the two constructions. First of all it is needed to specify the frozen complement for *"stellen"*, which contributes the essentials of the meaning of the whole phrase. Both `concept` and `lxm` slot are specified to obtain not only a noun with the same lexical string but also the right reading (there may be more). The syntactic constraints on the `instance` slot are the same as with "ordinary" direct objects, thus the whole phrase may undergo passivization, *"Antrag"* may be topicalized etc. which is correct with this particular support verb construction.

Secondly, this slot is also the repository of all information that is not present in the *"beantragen"* case (with exception of the `range` slot) as it pertains to the noun *"Antrag"*, e.g. particular kinds of reference (definite, deictic etc.) or modification.

The `actee` slot in Fig. 4 simply is a pointer to the `range` argument of *"Antrag"*. This is due to the fact, that the "thing that is applied for" is not only semantically but also syntactically an argument of the noun and has to obey the syntactic and word-order restrictions imposed by *"Antrag"*. Thus it is realized upon expanding the `args` of the NP with head *"Antrag"* and not at the verb level.

Idioms, that are more restricted w.r.t. passivization and modification can be represented as well. Consider, e.g., example (1), which neither can be passivized nor allows modification of *"Löffel"*. Precisely these two

```
((cat       lex-verb)
 (lxm       "abgeben")
 (concept   die)
 ;; --- inhibit passivization
 (reduction no)
 (args
   ((actor #(external np-ext))
   ;; --- "frozen" complement
    (actee #(external pp-int)
           ((lxm "Loeffel")
           ;; --- inhibit modification
            (args NONE)
            (reference definite))))))
```

Figure 5: Lexical Entry for *"den Löffel abgeben"* in FUF

possibilities are excluded in the lexical definition of the idiom in Fig. 5, modification of the main verb—which is allowed— is still possible as well as topicalization of the frozen complement or intervening negation particles (which would not be the case when representing the idiom as a fixed string in the lexicon).

5 Conclusion

The chosen approach of implementing a tactical generator based on FUF by emulating lexicon driven processing within the HPSG-style grammar has proven to be well suited for the unruly task of generating idioms. Since idioms are a primarily lexical phenomenon, the definition of argument structure in the lexicon is of great use. Subcategorizing for signs (as opposed to the `subcat` definition in [Pollard and Sag, 1994] which only constrains `synsem` values), as implemented in the generator, proved to be an essential advantage for representing idioms in the lexicon.

Thus the architecture of the generator fulfils the requirements necessary for the handling of idiomatic and collocational expressions: Frozen complements can be specified in the lexicon, even up to assigning particular lexemes to them. Furthermore they are represented in the same way as "ordinary" complements, thus case assignment, topicalization or passivization is possible using the standard syntactic mechanisms. Nevertheless, in case of idioms with very strict wording and little variation possibilities the lexical means are provided to inhibit the operations that are incompatible with the idiomatic reading.

References

[Elhadad, 1991] Elhadad, M. 1991. FUF: The Universal Unifier User Manual, Version 5.0. Technical report, Dept.of Computer Science, Columbia University.

[Heinz and Matiasek, 1994] Heinz, W. and J. Matiasek. 1994. Argument Structure and Case Assignment in German. In John Nerbonne, Klaus Netter, and Carl Pollard, editors, *German in Head-Driven Phrase Structure Grammar*. CSLI Publications, Stanford, pages 199–236.

[Kasper, 1989] Kasper, R. T. 1989. A flexible interface for linking applications to Penman's sentence generator. In *Proceedings of the DARPA Speech and Natural Language Workshop*, Philadelphia.

[Kay, 1979] Kay, Martin. 1979. Functional Grammar. In *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistics Society*. Berkeley Linguistics Society, Berkeley, CA.

[Krenn and Erbach, 1994] Krenn, B. and G. Erbach. 1994. Idioms and Support Verb Constructions. In John Nerbonne, Klaus Netter, and Carl Pollard, editors, *German in Head-Driven Phrase Structure Grammar*. CSLI Publications, Stanford, pages 265–296.

[Matiasek and Buchberger, 1995] Matiasek J.and E. Buchberger. 1995. A Tactical Generator for German Combining HPSG and FUF, in *Proceedings of the 5th European Workshop on Natural Langugage Generation* , Leiden, The Netherlands, May 20-22, 1995.

[Matthiessen, 1991] Matthiessen, C. 1991. Lexico(Grammatical) Choice in Text Generation. in Paris C.L., et al., editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, Kluwer, Dordrecht.

[Pollard and Sag, 1987] Pollard, C. and I. Sag. 1987. *Information-Based Syntax and Semantics, Vol. 1: Fundamentals*. CSLI Lecture Notes 13. CSLI, Stanford, CA.

[Pollard and Sag, 1994] Pollard, C. and I. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago and CSLI Publications, Stanford, CA.

[Trost, 1991] Trost, Harald. 1991. X2MORF: A Morphological Component Based on Augmented Two-Level Morphology. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, Sydney, Australia.

[Trost and Matiasek, 1994] Trost, Harald and Johannes Matiasek. 1994. Morphology with a Null-Interface. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, Kyoto, Japan, August 5-9.

[Zadrozny, 1992] Zadrozny Wlodek. 1992. On Compositional Semantics. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, Nantes, France.