A Proposed Standard for the Lexical Representation of Idioms

Jan Odijk
University of Utrecht
Trans 10
3512 JK Utrecht
The Netherlands
jan.odijk@let.uu.nl

Abstract
In this paper I first explain briefly the properties of one type of Multi-Word Expression (MWE), viz., flexible idioms, and how they are dealt with in the Rosetta machine translation system. Taking this as a starting point and generalizing beyond it, I argue that a standardized lexical representation for flexible idioms is not so straightforward. Nevertheless, I make a very concrete proposal for an actual standard for flexible idioms that is highly theory-neutral, and I show how it allows one to achieve a significant reduction of effort in reusing lexical entries for idioms.

1. Introduction

State-of-the art NLP systems do not deal adequately with large numbers of MWEs and this forms a major obstacle for the successful application of NLP technologies. (Sag *et al.* 2001) is titled: *Multiword expressions: a pain in the neck for NLP* and states that ``Multiword expressions are a key problem for the development of large-scale, linguistically sound natural language processing technology".

This problem can be overcome by having two ingredients: (1) an adequate method of handling each type of MWE in the grammar of the NLP system; (2) the availability of a large number of lexical entries for MWEs compatible with the requirements of the grammar.

The first ingredient has been the subject of a lot or research and has resulted in a wide variety of approaches in different grammatical frameworks and different implementations.[1] This paper focuses on the second ingredient for one type of MWE, viz. flexible idioms. I first explain briefly the properties of flexible idioms and how they are dealt with in the Rosetta machine translation system. I explore the possibilities for developing a standard for the lexical representations of these idioms. Though I argue that devising a standardized lexical representation for flexible idioms is not so straightforward, I nevertheless make a very concrete proposal for an actual standard for flexible idioms that is simple and highly theory-neutral, and I show how it allows one to achieve a significant reduction of effort by reusing lexical entries for flexible idioms created independently. I illustrate the method by means of a concrete example in which the method is applied. I introduce briefly one extension of the original method. I discuss various problems one might encounter when applying the method, and potential objections to the proposed method. I show that almost all these `problems' and `objections' are in fact virtues of the proposed method. Finally, I summarize the major conclusions.

2. Flexible Idioms

A principled and very powerful method for dealing with *flexible idioms* has been developed in the Rosetta project and implemented in the Rosetta system. We will briefly describe this method here. For further details we refer to (Rosetta 1994) and especially to (Schenk 1994).

Flexible idioms are called *flexible*[2] because next to a canonical order with contiguous elements (as in (1a)), it also allows other words to intervene between its components (as in (1b)), it allows permutations of its component words (as in (1c)), and combinations of permutations and intervention by other words not part of the idiom (as in (1d) :

(1) a. Hij heeft gisteren **de plaat gepoetst**
lit. 'He has yesterday the plate polished'
b. Ik dacht dat hij gisteren **de plaat** wilde **poetsen**
lit. 'I thought that he yesterday the plate wanted polish'
c. Hij **poetste de plaat**
lit. 'He polished the plate'
d. Hij **poetste** gisteren **de plaat**
lit. 'He polished yesterday the plate'

By assigning a flexible idiom the syntactic structure that it would have as a literal expression, it will participate in the syntax as a normal expression, and permutations, intrusions by other words or phrases, etc. can occur just as they can occur with these words in their literal interpretation.

Flexible idioms often have restrictions on their syntactic behavior additional to the ones on non-idiomatic constructions. Many of these restrictions can be predicted from general principles (given an adequate description of the idioms) and should therefore follow from the design of the grammar used in an NLP system (see (Schenk 1994) for one approach). Other restrictions on idioms cannot always be reduced to general grammatical properties or principles, and must be stipulated as idiosyncratic properties, e.g. passivization.

Furthermore, the grammar must of course differentiate an idiom from its literal counterpart. In the approach adopted in Rosetta, this imposes requirements on the lexical representation of idioms. A flexible idiom is described in the lexicon with a number of properties specific to idioms, in particular a syntactic structure, and a list of lexical item identifiers making up the idiom. The syntactic structure is not directly represented in the lexicon with the lexical item for the idiom. Instead, a unique name for (reference to) the syntactic structure, called *idiom pattern* in Rosetta, is specified. The syntactic structures themselves are derivation trees (D-trees).

The actual representation of some Dutch flexible idioms containing the verb *gaan* 'to go' in the Rosetta lexicon is given in Table 1. In the Rosetta lexicons, idioms are listed with the lexical entry for their head, but this is not essential in any way. The example shows the stem of the Dutch verb *gaan* 'go', which is *ga*, followed by a unique identifier for the syntactic item, called the *syntactic key* or *skey* ($s_aV_00_ga). Next, there are properties for two idioms, viz. *op de fles gaan* (lit. 'to go on the bottle', idiomatically 'to go bankrupt'), and *de pijp uitgaan* (lit. 'to go out of the pipe', idiomatically 'to die'). For each idiom the

following properties are specified: (1) a sequence of skeys for the non-head components of the idiom. For *de pijp uitgaan* these are $s_prep1286700 for *uit*, and $s_aN_00_pijp for *pijp*. The skey for the head is not in this list because the idiom is described as part of the lexical entry for the head. There is no skey for the article *de* because an article is introduced syncategorematically (i.e. it is introduced by a rule that it is not an argument of). Finally, the order of the skeys in the list is crucial: it must match the order of the elements in the D-tree for the idiom (see below); (2) idiom pattern (*vpid87*). A simplified version of the syntactic structure associated to this idiom pattern (literally: `to go out of the pipe', idiomatically: `to die') is represented in Fig. 1; (3) an skey for the idiom as a syntactic unit (*$s_id_depijpuitgaan*); (4) a unique identifier (mkey) for each meaning of the idiom. The idiom *de pijp uitgaan* has been assigned the mkey *$m_id_depijpuitgaan*; (5) a meaning description for each meaning ("*dood gaan*", `to die').

Lexical item element	**Explanation**
Ga :$s_aV_00_ga	Stem Skey
:<$s_prep1286400 $s_aN_00_fles> [vpid87] $s_id_opdeflesgaan $m_id_opdeflesgaan "failliet gaan"	skeys of idiom parts idiom pattern idiom skey (lit. `go on the bottle') idiom mkey idiom meaning description (`go bankrupt')
< $s_prep1286700 $s_aN_00_pijp > [vpid30] $s_id_depijpuitgaan $m_id_depijpuitgaan "dood gaan"	skeys of idiom parts idiom pattern idiom skey (lit. `go out of the pipe') idiom mkey idiom meaning description (`to die')
{ ... }	syntactic properties and the meanings of *gaan* (not specified here)

Table 1: Representation of some flexible idioms in the Rosetta Dutch lexicon

The D-tree associated to an skey for an idiom is used to generate the complex structure for the idiom. Once this structure has been created, it is subject to all the normal rules of the grammar, and will participate normally in syntactic processes.

3. A Standard for Idioms?

In the preceding sections we have discussed flexible idioms, one of the most difficult types of MWEs to deal with in NLP systems. The issue I would like to address in this section is whether we can make a proposal for a standard representation of these idioms that has at least some initial likelihood as a successful candidate *de facto* standard.

Good candidates for standards have to meet a lot of requirements, but I want to focus on two of them in particular, viz. high-degree of theory-independence, and technical feasibility. If we compare the treatment of the flexible idioms in Rosetta, and investigate the

possibility of a standard for this type of idioms meeting these requirements, it is easy to see that this is not so straightforward. Let me repeat what ingredients were necessary to adequately describe flexible idioms in Rosetta: (1) a syntactic structure for the idiom; (2) unique identification of the idiom components; (3) listing of the idiom components compatible with the syntactic structure. Can we propose reasonable standards for each of these aspects? A standard for the representation of the syntactic structure of idioms has been proposed in the ISLE and XMELLT projects,[3] but it appears to me to be highly ambitious and to have little chance of being successful. The syntactic structure assigned to an idiom is

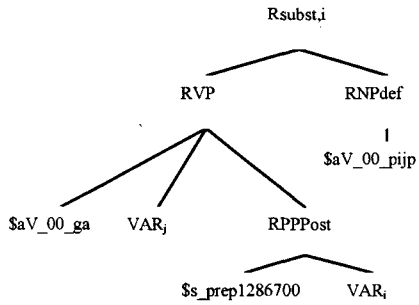

Figure 1: Rosetta D-tree for the idiom *de pijp uitgaan* (simplified)

highly theory-dependent. Not only are the structures assigned to idioms highly theory-bound, within one theory there will be many differences from implementation to implementation. A standard should perhaps abstract from such differences, but it is not clear that it will then still be possible to transform the standard representation into the representation of a specific system, which is one of the major reasons for wanting to have a standard to begin with. Finally, the representation of syntactic structures, typically tree structures with nodes and labels on the nodes where the labels can be quite complex attribute-value matrices, is very complex and very difficult to create and maintain. Typically, these representations are rather unstable: they have to change while the system for dealing with single words is still under development or even when it is in maintenance mode. In fact, I believe it is impossible to create and maintain such representations if one is not aided by a concrete implemented system.

If we look at the second aspect, the unique identification of the idiom components, prospects are not very good either. Note that it is generally not enough to identify the idiom components by specifying a string: in the Rosetta system skeys were used for the proper identification. It is highly unlikely that a standard and commonly accepted lexicon can be created so that every developer can find a unique reference to each idiom component there. Furthermore, every lexicon is incomplete, so a generally accepted method should exist to add new lexical items in such a way that they immediately become available to all other researchers. All of this seems very unlikely to me.

Third, let us consider the listing of the idiom-components: these must be listed in a way that is compatible with the syntactic structure assigned to the idiom. In the Rosetta

system there are many highly theory and implementation specific aspects to this: in particular the order and the presence of certain components (e.g. articles) of the idiom components is highly theory and implementation specific. Despite these problems, I believe a proposal can be worked out that has a chance of succeeding, provided that the problem of the unique identification of the idiom components is solved (see below for a concrete proposal).

Finally, let us assume, for the sake of the argument, that there is a standard representation of the structure of idioms, that we have solved the problem of uniquely identifying the idiom components, and their listing in the idiom component list: in this case, how would this standard be used? One major use of the standard representations should be an automatic transformation into the system-specific representations for a wide range of systems. But will that be possible in an easy way? I doubt that very much. Even if an automated procedure is possible, it will be a highly complex one and will require a lot of effort in implementing. It is doubtful that the creation and use of such a procedure will be more efficient than having the candidate structures generated by the system itself combined with manual selection on the basis of information obtained from the standard structure.

My overall conclusion from these considerations is that it is very unlikely that a theory-independent and technically simple standard specifying how the structure of flexible idioms should be described can be devised at all.

4. A Proposed Standard for Idioms!

Despite the conclusions drawn in the preceding section, I will propose a standard for idioms in this section. The proposed standard covers flexible idioms and avoids most if not all of the problems brought up. The central idea behind the proposal is that the proposed standard does not prescribe the structure of an idiom, but backs off to a slightly weaker position, viz. it specifies which idioms have the same structure. In short, it requires that equivalence classes of idioms are created based on whether they have the same structure. Having these equivalence classes reduces the problem of assigning a concrete structure and properties to an idiom to one instance of the class. And for this problem, we make a concrete proposal in which the relevant information is to a large extent generated by the concrete systems in which the idioms will be used.

4.1 The proposed standard

In order to get concrete, I propose that an idiom description should consist of the following parts: (1) an idiom pattern, i.e. an identifier that uniquely identifies the structure of the idiom. The equivalence classes are defined with the help of these idiom patterns: idioms with the same idiom pattern belong to the same equivalence class; (2) a list of idiom components (ICL). This takes the form of a sequence of strings, each string representing the lexicon citation form of each idiom component. The list must contain a citation form for each idiom component (so articles, left out in the Rosetta approach, should be included). As to order, the proposal leaves the order free, but only imposes the requirement that the same order is used for each instance in the same equivalence class; (3) an example sentence that contains the idiom. The structure of the example sentence should be identical for each example sentence within the same equivalence class.

Next to the idiom description, we need a description of the idiom patterns. This is a list of idiom pattern descriptions, where each idiom pattern description consists of two parts: (1) an idiom pattern, and (2) comments, i.e. free text, in which it is clarified why this idiom pattern is distinguished from others and further indications are given to avoid any possible ambiguities as to the nature of the idiom structure. It is even possible to supply a more or less formalized (partial) syntactic structure here, but the information in this field will be used by human beings and not be interpreted automatically.

This concludes the description of the proposed standard for the lexical representation of idioms. In order to illustrate how it can function as a useful standard, I will first give an example that follows this standard. Then I will describe a procedure to convert these descriptions into a system-specific description, and I will illustrate this procedure by deriving Rosetta-specific structures for these examples.

Table 2 shows 3 instances of the same idiom equivalence class from Dutch, and gives a description of the idiom pattern used to define this equivalence class

Idiom pattern	**Idiom components**	**Example**
Idiomp1	De pijp uit gaan	Hij is de pijp uitgegaan
Idiomp1	het schip in gaan	Hij is het schip ingegaan
Idiomp1	De boot in gaan	Hij is de boot ingegaan
Idiomp1	Verb taking a subject and a directional PP headed by a postposition and with an NP complement consisting of a determiner and a singular noun.	

Table 2: Three instances of idiom equivalence class *idiomp1* and the description of the equivalence class.

4.2 The conversion procedure

The procedure to convert a class of idiom descriptions made in accordance with the standard proposed into a class of idiom descriptions for a specific system consists of two parts: a manual part, and an automatic part. The manual part has to be carried out once for each idiom pattern, and requires human expertise of the language, of linguistics, and of the system into which the conversion is to be carried out. The automatic part has to be applied to all instances of each equivalence class.

The manual part of the conversion procedure for a given idiom pattern *P* consists of 5 steps: (1) select an example sentence for idiom pattern *P*, and have it parsed by the system, yielding the reference parse; (2) define a transformation ('parse transformation', PT) to turn the parse structure into the idiom structure; (3) use the result of the parse to determine the unique identifiers of the lexical items used in the idiom ('Idiom Component Identifier List', ICIL); (4) use the structure resulting from the parse to define a transformation to remove and/or reorder lexical items in the idiom component list ('Idiom Component List Transformation', ICLT); (5) apply the ICLT and make sure that the citation form of each item in the ICIL equals the corresponding element on the transformed ICL.

The automatic part of the conversion procedure is applied to each instance of the equivalence class defined by idiom pattern *P*, and also consists of 5 steps: (1) Parse the

example sentence of the idiom and check that it is identical to the reference parse, except for the lexical items; (2) use the PT to turn the parse tree into the structure of the idiom; (3) select the ICIL from the parse tree, in order; (4) apply the ICLT to the ICL; (5) check that the citation form of each item on the ICIL equals the corresponding element on the transformed ICL.

4.3 Illustration

I will illustrate the procedure by applying it to the examples given above and deriving the representation required in the Rosetta system. We first apply the manual part for the idiom pattern *idiompl*: (1) we select the example sentence *Hij is de pijp uitgegaan.* Parsing it by the Rosetta system yields the syntactic D-tree of Fig. 2; (2) The PT is simple: delete everything above the node containing the (parameterized) rule *Rsubst,i.* Applying PT to the tree of Fig. 2 yields the tree of Fig. 1; (3) given the resulting tree, the ICIL consists of $aV_00_ga, $s_prep1286700, and $aV_00_pijp, in this order; (4) the citation forms listed on the ICL (*de pijp uit gaan*) can be brought in correspondence with the ICIL by applying the transformation 1 2 3 4 → 4 3 2 (i.e. delete the first element and reverse the remaining list). Applying this transformation turns *de pijp uit gaan* into *gaan uit pijp*; (5) the citation forms of the skeys correspond to the elements on the transformed ICL (cf= citation form of): cf($aV_00_ga) = *gaan*, cf($s_prep1286700) = *uit*, cf($aN_00_pijp) = *pijp.*

In this way we have obtained a procedure to convert idioms of idiom pattern *idiompl* represented in the standard format proposed into the structure required in the Rosetta system.

The automatic part is applied to each instance of the equivalence class. As illustration, we apply it to the idiom *het schip ingaan* 'to have bad luck'. We follow the steps described above: (1) parsing the example sentence *hij is het schip ingegaan* indeed leads to a syntactic D-tree that is identical to the reference parse, except for the skeys; (2) the PT turns it into the D-tree of Fig. 1, with $s_prep1286700 replaced by $s_prep1286800, and with $aN_00_pijp replaced by $aN_00_schip; (3) the skeys for the lexical items in this D-tree are $aV_00_ga, $s_prep1286800, and $aN_00_schip, in this order; (4) the idiom component list transformation applied to the ICL *het schip in gaan* yields *gaan in schip*; (5) the citation form of each item in the ICIL equals the corresponding element on the

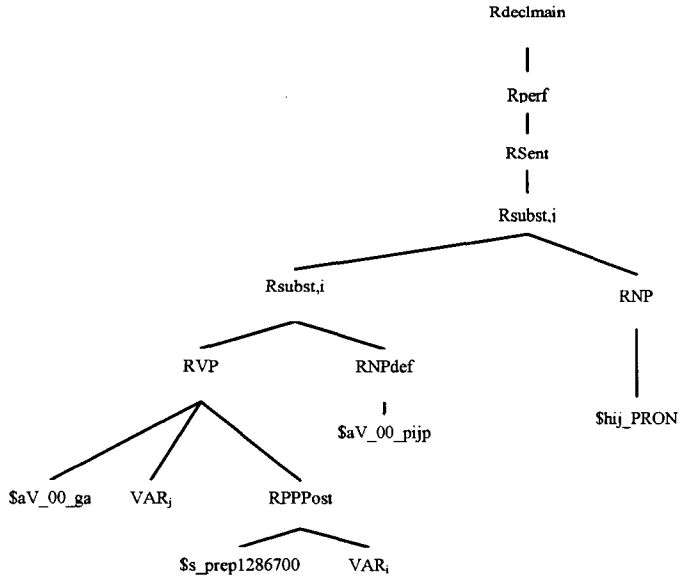

Figure 2: Syntactic D-tree for *Hij is de pijp uitgegaan*

transformed ICL: cf($aV_00_ga) = *gaan*, cf($s_prep1286800) = *in*, cf($aN_00_schip) = *schip*.

Hence, we have derived the representation for the idiom *het schip ingaan* in Rosetta in a fully automatic manner: (1) the syntactic D-tree; (2) the skeys used in the idiom, (3) in an order compatible with the syntactic D-tree. It is trivial to convert this result in the precise format required in Rosetta

5. Possible problems and objections

In the preceding section we illustrated the procedure to derive a system-specific representation for idioms from the proposed standard representation. But we illustrated an idealized case only. There are many steps in the procedure that could yield other results than the ones illustrated. In this section we will discuss these.

The first step is to parse the example sentence illustrating the idiom pattern. The resulting parse tree is then used in the further steps. Of course, in general, a system will yield a (possibly empty) set of parse trees, and there is no guarantee that the correct parse tree for this idiom is included in this set. If the correct parse tree is not included in the resulting set of parse trees, it usually means that the system's lexicon and/or grammar are incorrect or incomplete. In such a case, it is actually a virtue that one is pointed out that the system cannot handle this idiom: it makes no sense to add it if it cannot lead to a correct parse anyway. The remedy is simple: extend the system's lexicon and/or grammar so that it does yield a correct parse.

One case requires special mentioning: though idioms generally have regular syntactic structures, many idioms use structures only allowed in idioms but not in non-idiomatic constructions. Examples are the use of singular count nouns in determinerless NPs (e.g. English *he was afraid of losing face*), the use of inalienable possession constructions in Dutch, examples such as Dutch ***ten tijde van*** 'at the time of' (containing the fossilized portmanteau word *ten* and the *e*-form of the noun *tijd*, both only used in idioms), etc. If such structures recur in many idioms they can be dealt with by including *minor rules* in one's system to describe such structures: minor rules are rules that only can be used to form idiomatic structures. The system should be used in a mode that allows minor rules to be applied for non-idiomatic structures as well when applying the method.

The set of parses may contain the correct parse but also other parses. In the automatic part, selection of the correct parse is done automatically by comparison with the reference parse. In the manual part, however, the correct one has to be selected manually on the basis of the idiom-pattern description and other linguistic and system-specific knowledge.

In the final step, there might be a mismatch between the citation form of the idiom component list and the one generated on the basis of the uniquely identified lexical item, especially for variable elements of the idiom. This will be detected automatically, and again, this is a virtue of the approach.[4]

It may also be the case that the developer, giving his/her knowledge of the system and the idiom pattern description, concludes that the current pattern collapses idioms in a single equivalence class while the specific system requires a further subdivision. Though this complicates matters, it cannot be an argument against the method proposed here. What we see here is that the proposed method is not completely theory-neutral. However, the same problem would also (in fact: much more often) arise in a proposal that describes how idiom structures look like.

6. Parameterization

Several extensions and improvements of the proposed standard are possible. In this section I briefly mention one. It extends the method with parameters. Lack of space prevents me from fully elaborating, formalizing and illustrating this proposal or discussing others.

A concrete example may help illustrate the use of parameters. Idioms can contain nouns. In Dutch, nouns can be singular (sg) or plural (pl), and positive (pos) or diminutive (dim). In the original proposal a different equivalence class would be needed for each of these 4 cases (and even more if more than one noun occurs in a single idiom). By introducing 2 parameters for nouns (sg/pl, pos/dim), it is possible to group these 4 equivalence classes into a single equivalence superclass, and to have a single PT for this superclass, which however is parameterized for the properties of the noun (sg/pl; pos/dim).

The extension with parameters introduces a little more theory and implementation specificity to the method, but it does so in a safe way: NLP systems that can make use of these parameters will profit from it, while systems that cannot make use of these parameters are not harmed since the original equivalence classes can still be identified. For the example given above the theory /implementation dependency that is introduced is that properties such as sg/pl and pos/dim on a noun are dealt with by rules applying to just the noun. It can be expected that many different grammatical frameworks share this assumption.

The extension contributes to reducing the number of equivalence classes and increasing the number of members within equivalence classes. It will therefore reduce the number of idioms that have to be dealt with manually and increase the number of idioms that can be incorporated into an NLP system in a fully automatic manner. This is important because the method proposed here categorizes flexible idioms into equivalence classes. The successfulness of this method will therefore depend on (1) how many different equivalence classes must be distinguished (the less the better), and (2) how many instances each equivalence class contains (the more the better).

We carried out measurements on two databases of idioms to determine this. The first database is a small database of 893 Dutch idioms (Dutch MiniDB) categorized into parameterized equivalence classes. The second database is the SAID database (Kuiper *et al.* 2003), in which we approximate such a classification by assuming that the delexicalized syntactic structures of this database correspond to parameterized equivalence classes. Table 3 presents the major findings of our measurements. The result, though not definitive, is promising. It means, e.g., that 80% (or 11,773) of the idioms in the SAID database can be dealt with by just 481 equivalence classes.

	SAID		Dutch MiniDB	
Cov.	#idioms	#patterns	#idioms	#patterns
50%	7383	28	449	21
60%	8853	54	539	36
70%	10304	140	628	59
80%	11773	481	716	98
85%	12509	908	760	134
90%	13245	1644	804	178
95%	13981	2380	849	223
100%	14716	3116	893	267

Table 3: Coverage of idiom patterns in two idiom databases

7. Conclusions

In this paper I have analyzed the properties of flexible idioms and how they are dealt with in the Rosetta machine translation system. Based on the analysis of the requirements, and generalizing beyond it, I have made a very concrete proposal for a standard for the lexical representations of these idioms. This proposed standard is very simple from a technical and linguistic point of view, it is highly theory-neutral, and it could be an important technique to allow for maximal reuse of lexical entries for idioms in many systems that may differ widely in terms of their theoretical basis, their actual implementation, and their treatment of idioms. Its technical simplicity and high theory independence also offers prospects of bridging the gap between traditional lexicographers and NLP developers.

I have not discussed many important classes of MWEs at all: fixed idioms, semi-flexible idioms, support verb constructions, lexical collocations, etc. However, I am convinced that the central idea behind the current proposal can also be applied to these types of MWEs. If correct, the current proposal would be a proposal that has the potential to be an all-encompassing proposal for all types of MWEs.

I have investigated the major potential stumbling block for the current proposal: if the number of different equivalence classes is very high, and the number of members of one equivalence class is low, then not much reduction of effort will be obtained. I concluded that this is not a stumbling block for the method: large coverage can be obtained with relatively little effort, but obtaining full coverage still requires a significant effort.

Acknowledgements

Part of the research reported on here was carried out in the context of the EC-funded ISLE project (http://www.ilc.pi.cnr.it/EAGLES96/isle/ISLE_Home_Page.htm).

Endnotes

1. Examples include the method developed in the context of the compositional grammars of the Rosetta approach of compositional translation (see (Schenk 1986; Schenk 1989; Schenk 1992; Schenk 1994; Landsbergen *et al.*1989; Rosetta 1994); in Government-Binding Theory-based implementations, e.g. (Wehrli 1998); in the context of Tree Adjoining Grammars (TAG): (Abeillé & Schabes 1989; Abeillé 1995); in the context of HPSG: (Riehemann 1997; Richter & Sailer 2002; Sailer & Richter 2002a; Sailer & Richter 2002b; Sailer 2002; Soehn 2003; Soehn & Sailer 2003); in the context of finite state techniques: (Breidt & Segond 1995a; Breidt & Segond 1995b).
2. Flexible idioms contrast with *fixed* idioms that consist of sequences of invariable words (with the possible exception of inflection at one edge) such as *ad hoc, Kuala Lumpur*, etc., and with (what I call) *semi-flexible* idioms, which allow inflection on each word but no permutations or intrusions by other elements. See (Odijk 2003).
3. Mainly for support verb constructions, see http://www.cs.vassar.edu/~ide/XMELLT.html, http://www.ilc.pi.cnr.it/EAGLES96/isle/ISLE_Home_Page.htm, and for compounds.
4. In addition, the citation form alone might not disambiguate sufficiently, e.g. for verbs that are partially but not fully conjugated in the same way, e.g. English *hang/hanged/hanged* v. *hang/hung/hung*. Such problems can be avoided by carefully selecting the examples sentences.

References

Abeillé, A. & Schabes,Y., 1989. 'Parsing Idioms in Lexicalized TAGs'. In Proceedings *of the European ACL*, pages 1-9, Manchester.

Abeillé, A., 1995. 'The Flexibility of French Idioms: A Representation with Lexicalised Tree Adjoining Grammar'. In Everaert, M. *et al.* (eds.), *Idioms: Structural and Psychological Perspectives*, chapter 1.,Hilldale, New Jersey/Hove, UK: Lawrence Erlbaum Associates.

Breidt L., & Segond, F., 1995a. 'Compréhension Automatique des Expressions à Mots Multiples en Français et en Allemand'. In *Quatrièmes journées scientifique de Lyon, Lexicomatique et Dictionnairiques*.

Breidt, L. & Segond, F., 1995b. 'Idarex: Formal Description of German and French Multi-word Expressions with Finite-State Technology'. *MLTT*, 022:1036--1040, November.

Kuiper, K., *et al.*, 2003. *SAID: A Syntactically Annotated Idiom Dataset.* LDC2003T10, Pennsylvania: Linguistic Data Consortium.

Landsbergen, J. *et al.*, 1989. 'The Power of Compositional Translation'. In *Literary and Linguistic Computing*, 4(3):191-199.

Odijk, J., 2003. 'Towards a Standard for Multi-Word Expressions'. ISLE Project Report, http://lingue.ilc.cnr.it/EAGLES96/isle/clwg_doc/ISLE_D6.1.zip, February.

Richter, F. & Sailer, M., 2002. 'Cranberry Words in Formal Grammar'. In Beyssade, C. *et al.*, *Empirical Issues in Formal Syntax and Semantics*, volume 4. Paris: Presses Universitaires de Paris-Sorbonne.

Riehemann, S., 1997. 'Idiomatic Constructions in HPSG'. Presented at the 4th International Conference on HPSG.

Rosetta, M.T., 1994. *Compositional Translation*, volume 273 of *Kluwer International Series in Engineering and Computer Science (Natural Language Processing and Machine Translation)*. Dordrecht: Kluwer Academic Publishers.

Sag, I., *et al.* 2001. 'Multiword Expressions: A Pain in the Neck for NLP'. *LinGO Working Paper*, (2001-03). http://lingo.stanford.edu/csli/pubs/WP-2001-03.ps.gz.

Sailer, M. & Richter, F., 2002a. 'Collocations and the Representation of Polarity'. In Alberti, G. *et al.* (eds.), *Proceedings of 7th Symposium on Logic and Language*, pages 129--138, Pécs, 26-29 August.

Sailer, M. & Richter, F., 2002b. 'Not for Love or Money: Collocations'. In Jäger, G. *et al.*, (eds.), *Proceedings of Formal Grammar 2002*, pages 149-160.

Sailer, M., 2002. 'The German Incredulity Response Construction and the Hierarchical Organization of Constructions'. Material to a talk presented at 2nd International Conference on Construction Grammar, Helsinki, 6-8 September.

Schenk, A., 1986. 'Idioms in the Rosetta Machine Translation System'. In *Proceedings of the 11th Conference on Computational Linguistics*, Bonn.

Schenk, A., 1989. 'The Formation of Idiomatic Structures'. In Everaert. M. and Van der Linden, E.-J. (eds.), *Proceedings of the First Tilburg Workshop on Idioms*, pages 145-158. ITK proceedings, Tilburg University.

Schenk, A., 1992. 'The Syntactic Behaviour of Idioms'. In Everaert, M. *et al.* (eds.) *Proceedings of Idioms, International Conference on Idioms*, Tilburg, The Netherlands, 2-4 September 1992, pages 97--110. ITK, Tilburg University.

Schenk, A., 1994. *Idioms and Collocations in Compositional Grammars.* Ph.D.-thesis, University of Utrecht.

Soehn, J.-P. & Sailer, M., 2003. 'At First Blush on Tenterhooks. About Selectional Restrictions Imposed by Nonheads'. In Jäger, G. *et al.* (eds.), *Proceedings of Formal Grammar 2003*, pages 149-161.

Soehn, J.-P., 2003. *Von Geisterhand zu Potte Gekommen. Eine HPSG-Analyse von PPs mit Unikaler Komponente.* Magisterarbeit, Universität Tübingen, Seminar für Sprachwissenschaft.

Wehrli, E., 1998. 'Translating Idioms'. In *Proceedings of COLING-ACL '98*, volume 2, pages 1388-1392, Montreal, Canada.