Towards a standard for Multi-Word Expressions

Jan Odijk

University of Utrecht, UiL-OTS, Trans 10, 3512 JK Utrecht, The Netherlands
jan.odijk@let.uu.nl,
WWW home page: http://let.uil-ots.nl

Abstract. In this paper I first explain briefly the properties of three types of Multi-Word Expressions (MWEs) of increasing complexity and how they are dealt with in two different machine translation systems. I explore the possibilities for developing a standard for the lexical representations of these MWEs. For the simplest type of MWE, a standard is easily devised, but for the more complex (flexible) types of MWEs the matter is not so straightforward. Nevertheless, I make a very concrete proposal for an actual standard for flexible MWEs that is highly theory-neutral, and show how it allows one to achieve a significant reduction of effort when one wants lexical entries for flexible MWEs created in one system to be reused in a totally different system

1 Introduction

In this paper I first explain briefly the properties of three types of Multi-Word Expressions (MWEs) of increasing complexity and how they are dealt with in two different machine translation (MT) systems. I explore the possibilities for developing a standard for the lexical representations of these MWEs. I state two important properties that proposed standards must meet in order to have a chance of being successful: technical simplicity, and independence of specific theories and implementations.

For the simplest type of MWE, a standard is easily devised, but for the more complex (flexible) types of MWEs the matter is not so straightforward. Nevertheless, I make a very concrete proposal for an actual standard for flexible MWEs that is simple and highly theory-neutral, and I show how it allows one to achieve a significant reduction of effort in one system by reusing lexical entries for flexible MWEs created in a totally different system

The paper is organized as follows. I start with giving some essential (but very minimal) background on the two machine translation systems discussed in this paper: the Rosetta MT system (section 1.1) and the Globalink MT system (section 1.2). Next, I describe the three different types of MWEs dealt with in this paper: fixed MWEs (section 2), semi-flexible MWEs (section 3), and flexible MWEs (section 4). I then discuss two requirements that good proposed standards must meet in order to have a chance to be successful at all, and show that it is not so easy to meet these requirements for a standard that prescribes how the structure of an MWE should be described (section 5). In section 6,

however, I make a concrete proposal for a standard for MWEs that avoids all of the problems pointed out in the preceding section. I illustrate the method by means of a concrete example in which the method is applied (section 6.3). Then I discuss various problems one might encounter when applying the method, and potential objections to the proposed method (section 6.4). I show that almost all these 'problems' and 'objections' are in fact virtues of the proposed method. Only one objection still has the potential of making the proposed standard less useful: whether this is the case is to be investigated, but if it is, it will, in my view, be an obstacle to any proposed standard in the domain of MWEs. The conclusions are summarized in section 7.

1.1 The Rosetta MT system

In this section I very briefly describe some of the crucial properties of the approach adopted in the Rosetta MT system that are relevant to the topic of this paper. The Rosetta MT system is a research prototype translating between Dutch, English and Spanish. It was developed in the Rosetta project carried out at the Philips Research Laboratories. For an extensive description of the approach and the system developed, I refer to Rosetta (1994).

The grammars of the Rosetta system are a special kind of *compositional* grammars called *M-grammars*. A compositional grammar consists of *basic expressions* and *rules*. One can derive utterances by recursively applying rules, initially to basic expressions. In order to be able to deal adequately with the complexities of natural language, rules apply to (one or multiple) syntactic trees called *S-trees* (short for surface trees) that encode syntactic categories, attribute-value pairs, linear order, and constituent structure. Basic expressions are a special kind of S-tree called a *lexical S-tree*. The way an utterance is derived can be represented by recording which rules must be applied, and to which arguments, initially basic expressions, in a syntactic derivation tree or *syntactic D-tree*.

Compositional grammars are designed in such a way that the principle of *Compositionality of Meaning* holds. This principle can be stated as follows:

(1) **Compositionality of Meaning**
The meaning of an expression is a function of the meaning of its parts and the way they are combined

This is achieved in compositional grammars because both basic expressions and rules have a meaning. Therefore, the meaning of an utterance can be derived in parallel with the syntactic derivation, and the way the meaning is computed can be represented in a *semantic D-tree* consisting of unique identifiers for the meanings of the rules and the meanings of the basic expressions.

The principle behind the method used to translate with M-grammars is called the principle of *Compositionality of Translation*:

(2) **Compositionality of Translation**
Two expressions are each other's translation if they are built up from parts which are each other's translation, by means of rules which are each others translation

Having the same meaning is a necessary condition for translational equivalence, but factors additional to meaning (e.g. stylistic) might be relevant in determining translational equivalence.

Compositionality of Translation is achieved by attuning M-grammars G1 and G2 of two languages, as follows:

- For each basic expression in grammar G1 there must be at least one basic expression in G2 that is translationally equivalent
- For each rule in grammar G1 there must be at least one rule in G2 that is translationally equivalent

Such attuned grammars are called *isomorphic grammars*. For each derivation tree in G1 there will be an isomorphic derivation tree in G2 that is translationally equivalent, and if the grammars satisfy certain additional restrictions, one can derive translationally equivalent utterances from G1 and G2 in parallel.

I have discussed the basic properties of the method of compositional translation only by speaking of the grammars (i.e. abstract characterizations of the language) used in this method. M-grammars satisfy a number of additional conditions that make them suitable to be used in actual analysis and generation of utterances. These, however, are less relevant for the topic of this paper, and I will not discuss them here. For details, I refer to Rosetta (1994).

A property that follows immediately from the approach and that is crucial for the treatment of MWEs is that an MWE that has a non-compositional meaning must be treated as a basic expression in the grammar: The design does not allow multiple expressions to map to one meaning (and it sharply contrasts here with other approaches, e.g. the Globalink approach). The treatment of MWEs in Rosetta as described below will reflect this.

1.2 The Globalink MT system

The Globalink MT system[1] is a transfer-based MT system. It has a grammar used in analysis and generation, but not for analyzing MWEs. The resolution of MWEs is carried out in transfer, where multiple lexical items may map to a single lexical item in the target language, and vice-versa.

2 Fixed MWEs

The simplest types of MWEs are *fixed* MWEs. *Fixed MWEs* in Rosetta consist of a sequence of words where

- the individual words occur in a fixed order

[1] The Globalink MT technology is owned by Bowne. I base the description given here on Odijk (2000), which supplies a description of the properties of Globalink lexicons that was made in the context of the ISLE project when this technology was owned by Lernout & Hauspie Speech Products.

- the individual words are always contiguous (no other elements can intervene)
- there is no variation in lexical item choice
- there is no inflection or only inflection at one edge[2]

Typical examples are fixed expressions and foreign geographic and other names that consist of multiple words:

(3) a. ad hoc 'ad hoc', stante pede 'stante pede', ter plaatse 'on the spot', by and large
b. Hong Kong, Kuala Lumpur, New York, San Francisco

but certain compounds and phrases in languages with little or no inflection (e.g English) could be dealt with as fixed MWEs as well:[3]

(4) credit card, travel agency, real estate agency

Examples of MWEs that cannot be treated as fixed MWEs are given in (5):

(5) a. (En.) mother-in-law
b. (It.) carta telefonica 'telephone card'
c. (Dutch) de plaat poetsen (lit. 'to polish the plate', 'to bolt')

The English example (5a) cannot be dealt with as a fixed MWE because it has internal inflection (cf. *mother***s**-*in-law*). The Italian example (5b) cannot be treated as such because each of its component words inflects (cf. *cart***e** *telefonic***he**). The Dutch example (5c) cannot be treated as a fixed MWE because next to a canonical order with contiguous elements (as in (6a)), it also allows other words to intervene between its components (as in (6b)), it allows permutations of its component words (as in (6c)), and combinations of permutations and intervention by other words not part of the MWE (as in (6d):

(6) 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 though 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'

[2] There is no principled reason to exclude inflection in the middle, but it is technically easier to deal with if it is excluded and it avoids the use of prefixes and suffixes as infixes, so that the morphological component can remain simple, at least in languages that do not have infixation but probably also in languages with infixation.

[3] That might not be a very principled approach, but it can be very convenient in the development of actual systems.

Fixed MWEs are treated in the Rosetta lexicons as normal lexicon entries that require one and allow more than one contiguous spaces in their orthographic representation. In analysis, the incoming sequence of words is mapped onto a single lexical tree by morphology before it enters syntax. This makes it very suitable for dealing with MWEs for which the internal syntax is unclear or irregular relative to the syntax of the system (such as *ad hoc*, *by and large*, Dutch *op en top* 'fully', etc.).

3 Semi-Flexible MWEs

The Globalink system allows for (what I will call) *semi-flexible MWEs*. In this type of MWE,

- the component words have to occur in a fixed order
- the component words have to be contiguous (words that are not part of the MWE cannot intervene)
- more than one part can inflect

Typical examples of such MWEs are given in (7):

(7) a. (En.) House of Representatives
b. (Sp.) patatas fritas
c. (Fr.) calculateur analogique
d. (Fr.) résistant aux acides
e. (It.) carta telefonica

None of these examples can be dealt with as a fixed MWE because of the internal or multiple inflection (cf. *House**s** of Representatives, patata**s** frit**as**, calculateur**s** analogique**s**, résistant**e** aux acides, cart**e** telefonic**he***)

Semi-flexible MWEs are represented in the Globalink lexicon (Lexicon Interchange Format, or LIF) as a sequence of the component words. The head of the phrase is marked as such, and each component word that can be inflected is represented by its canonical form and marked by an asterisk. Typical representations are:

(8) a. (En.) **House*** of Representatives
b. (Sp.) **patata*** frito*
c. (Fr.) **calculateur*** analogique*
d. (Fr.) **résistant*** aux acides
e. (It.) **carta*** telefonico*

where the head is marked by putting it in bold face.

In analysis, each word of the incoming sequence of words passes through morphology as an independent word, resulting in a canonical form and a morphological characterization for each word. They are subject to the normal rules of syntax (which checks on the legitimateness of the combination and checks agreement where applicable) If a sequence of canonical forms and or surface forms is

listed as flexible MWE, it is mapped as whole to its translation equivalent in the target language.

In generation, the MWE is introduced in transfer with multiple components, and an indication of the head. It participates in the syntactic rules as if it were a normal combination of these components, and nothing special has to be done.

Semi-flexible MWEs can deal with slightly more complex constructions than fixed MWEs, but MWEs with irregular internal syntactic structure require a treatment in the system similar to the one described for fixed MWEs in the Rosetta system. This can be implemented by treating all MWEs without internal inflection in this way.

Semi-flexible MWEs do not constitute a separate class of MWEs in Rosetta: the only way to deal with them is to treat them as flexible MWEs, to be discussed in the next section.

4 Flexible MWEs

A principled and very powerful method for dealing with *flexible MWEs* 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 Schenk (1986), Schenk (1992), and Schenk (1994).

As we have seen above, a flexible MWE can not only occur in a canonical order with contiguous components, it also allows other words to intervene between its components, it allows its component words to occur in different orders, and combinations of permutations and intervention by other words not part of the MWE. For convenience, we repeat the example given above illustrating this:

(9) a. Hij heeft gisteren **de plaat gepoetst**
b. Ik dacht dat hij gisteren **de plaat** wilde **poetsen**
c. Hij **poetste de plaat**
d Hij **poetste** gisteren **de plaat**

In addition, certain flexible MWEs allow for (and require) controlled variation in lexical item choice, e.g. in idiomatic expressions containing bound anaphora such as *to lose one's temper*, where the possessive pronoun varies depending on the subject:

(10) a. *I* lost *my* temper
b. *You* lost *your* temper
c. **I* lost *your* temper
d. **You* lost *my* temper

Of course, not every flexible MWE allows all of these options, and not all permutations of the components of a flexible MWE are well-formed (e.g. one cannot have **Hij heeft gepoetst plaat de*). The way to account for the properties of flexible MWEs with regard to these phenomena is to assign to a flexible MWE the syntactic structure that it would have as a literal expression: it will then

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.

Adopting this approach for the Dutch MWE *de plaat poetsen* accounts immediately for (9b), where the verb *poetsen* participates in the formation of verbal clusters in the normal way, just as the expression under its literal interpretation. The examples (9c,d) are also accounted for by assigning the MWE a normal syntactic structure: it can then be subject to the rule of *Verb Second* in the normal way, just as under the literal interpretation. It also accounts for the ill-formedness of the example **Hij heeft gepoetst plaat de* given above, since this string is also ill-formed under the literal interpretation.

Flexible MWEs often have restrictions on their syntactic behavior additional to the ones on normal constructions. Part of these restrictions can be accounted for by the fact that the parts of MWEs often do not have a meaning of their own in the MWE (only the MWE as a whole has a meaning), and that these parts do not refer. These additional restrictions should follow from the design of the grammar and its treatment of idiomatic MWEs, and not be stipulated for each individual MWE. In the Rosetta system many restrictions on the syntactic behavior of MWEs or their components are dealt with in a systematic way by the design of the grammar (e.g. restrictions on modification, topicalization, rearrangements in the "Mittelfeld" in Dutch, pronominalization, etc.).

Other restrictions on MWEs cannot be reduced to general grammatical properties or principles, and must be stipulated as idiosyncratic properties of the MWE. For example, certain expressions can be passivized only under the literal interpretation but not under the idiomatic interpretation.

Furthermore, an MWE must of course be recognized as such and differentiated from its literal counterpart. So at some point in the grammar, the treatment of MWEs must differ from their literal counterparts.

In the approach adopted in Rosetta, this is dealt with as follows. A flexible MWE is described in the lexicon, but it has a number of properties specific to MWEs, in particular

- a syntactic structure
- a list of lexical items making up the MWE

The syntactic structure is not directly represented in the lexicon with the lexical item for the MWE. Instead, a unique name for (reference to) the syntactic structure is specified. This is done because the syntactic structures are quite complex and are shared by multiple MWEs. Using names for syntactic structures rather than the syntactic structures themselves with the lexical items reduces the effort to add MWEs that require a structure already used for other MWEs, it increases the consistency and makes it easier to maintain the lexicon. The names are called *idiom patterns* in the Rosetta system

The syntactic structures themselves are not S-trees but D-trees. The reasons for this are as follows: S-trees contain nodes with a lot of attribute-value pairs. It is very difficult to get all the values of the attributes correct by hand (and

some manual work is required even if the lexicographer is supported by the system). Second, the nature of the attributes, and especially their values, are rather unstable during development of the system: new attributes are added, existing attributes removed or changed, and especially their values regularly change or are extended during development.[4] D-trees are a much more stable part of the system, and D-trees are, in comparison to S-trees, relatively simple: most nodes have atomic labels for rule names, and only few nodes have attributes (rule parameters).

The actual representation of some Dutch flexible MWEs containing the verb *gaan* 'to go' in the Rosetta lexicon is given in Table 1.

Lexical item element	Explanation
Ga	stem
:$s_aV_00_ga	skey
:<$s_prep1286400 $s_aN_00_fles>	skeys of MWE parts
[vpid87]	MWE pattern
$s_id_opdeflesgaan	MWE skey (lit. 'go on the bottle')
$m_id_opdeflesgaan	MWE mkey
failliet gaan	MWE meaning description ('go bankrupt')
< $s_prep1286700 $s_aN_00_pijp >	skeys of MWE parts
[vpid30]	MWE pattern
$s_id_depijpuitgaan	MWE skey (lit. 'go out of the pipe')
$m_id_depijpuitgaan	MWE mkey
dood gaan	MWE meaning description ('to die')
{ ... }	syntactic properties and the meanings of *gaan*

Table 1. Representation of some flexible MWEs in the Rosetta Dutch lexicon

In the Rosetta lexicons, MWEs 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*, 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 MWEs, 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 MWE the following properties are specified:

- a sequence of skeys for the non-head components of the MWE. For *de pijp uitgaan* these are
  - $s_prep1286700, the skey for the postposition *uit*[5]

[4] Though one cannot start working on developing lexical entries for MWEs before a significant part of the syntax for single words has been developed and is reasonably stable, it is also unrealistic to expect that MWEs will only be added to a fully stable and unchanging system.

[5] The correct way to analyze *uit* in this MWE is as a separate postposition. The fact that it is written together with *gaan* as a single word in certain cases is due to normal and independently required rules of Dutch orthography.

- $s_aN_00_pijp, the skey for the noun *pijp*

The skey for the head is not in this list because the MWE is descibed as part of the lexical entry for the head, so the skey head can be added automatically. There is no skey for the article *de* because articles are 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 MWE (see below)

– MWE pattern (*vpid87*), i.e. a unique identifier for the D-tree for this MWE. A simplified version of the syntactic structure associated to this MWE pattern (literally: 'to go out of the pipe', idiomatically: 'to die') is represented in Fig. 1.
– an skey for the MWE as a syntactic unit (*$s_id_depijpuitgaan*)
– an mkey for each meaning of the MWE. The MWE *de pijp uitgaan* has only one meaning, which has been assigned the mkey *$m_id_depijpuitgaan*[6]
– a meaning description for each meaning (*dood gaan*, 'to die')

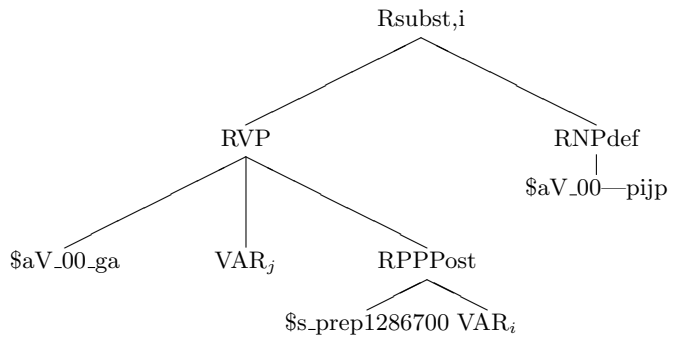


Fig. 1. Rosetta D-tree for the MWE *de pijp uitgaan* (simplified)

In analysis, the surface structure of an utterance created by the surface parser and represented in an S-tree, is subject to grammatical rules that check the well-formedness of the S-tree, and gradually modify and reduce this structure to end up with an S-tree in a canonical form (e.g effects of displacements such as Dutch Verb-second, verbal cluster formation, topicalization, etc. are undone), so that the argument structure of the utterance can be checked. Which rules are applied, and to which arguments, is recorded in the syntactic derivation tree for the utterance. At the point in the derivation where syntactic selectional restrictions are checked, it is also checked whether the structure can be analyzed as an MWE. To that end, it is checked whether the structure can be analyzed

[6] Several MWEs do have more than one meaning, e.g. Dutch *de pijp aan Maarten geven*, literally, 'to give the pipe to Maarten' can mean either 'to die' or 'to quit'.

in accordance with the D-tree of any MWE. If that is the case, the structure is transformed into a simplified structure in which the multiple parts of the MWE have been replaced by a single node, and further only the arguments of the MWE (if any) are present. The resulting structure is then subject to any further rules of the grammar, that, inter alia, check whether the right number of arguments is present: when this is the case, the full analysis of the utterance containing an MWE yields a derivation tree that is indistinguishable from derivation trees for utterances not containing any MWE.

In generation, the D-tree associated to an skey for an MWE is used to generated the complex structure for the MWE. Once this structure has been created, it is subject to all the normal rules of the grammar, and will participate normally in syntactic processes. The non-referential nature of the MWE's components will automatically account for several restrictions on the MWE's syntactic potential.

5 A Standard for MWEs?

In the preceding sections we have seen three types of MWEs of increasing complexity. The issue I would like to address in this section is whether we can make a proposal for a standard representation of these MWEs that has at least some initial likelihood as a successful candidate *de facto* standard.[7]

Good candidates for standards have to meet a lot of requirements, but I want to focus on two of them in particular:

- high-degree of theory-independence
- technical feasibility

If we first consider the fixed and semi-flexible MWEs, then the representations used, and their applicability, though certainly not completely theory-neutral, appear to me to be theory-neutral to a high degree, and technically very simple: the representations used are simple and can easily be transferred to representations required in completely different systems. So from the perspective of theoretical independence and technical feasibility, the methods of description used for fixed and semi-flexible MWEs are likely candidates to succeed as a *de facto* standard. Whether they will, will depend to what extent they will be accepted and really used by the community of NLP technology developers.

However, if we turn to the treatment of the flexible MWEs, the situation is completely different. Let me summarize what ingredients were necessary to adequately describe flexible MWEs in Rosetta:

- A syntactic structure for the MWE
- Unique identification of the MWE components
- Listing of the MWE components compatible with the syntactic structure

[7] I am more interested in whether they can be a potentially successful candidate for a *de facto* standard than in their viability as an official standard (e.g. approved by ISO, etc.). The latter can be important and can help turning a standard into a *de facto* standard, but the first is more important in my view.

Can we propose reasonable standards for each of these aspects? A standard for the representation of the syntactic structure of MWEs is proposed in the ISLE project, but it appears to me to be highly ambitious and to have little chance of being successful. The syntactic structure assigned to an MWE is highly theory-dependent. Not only are the structures assigned to MWEs 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 syste,, 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. Our experiences in the Rosetta project showed that maintenance of S-trees was practically impossible, but even to create and maintain the representation of MWEs by means of D-trees would have been very difficult if not impossible if we could not use the Rosetta-system itself as an aid.

If we look at the second aspect, the unique identification of the MWE components, prospects are not very good either. Note that it is generally not enough to identify the MWE 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 MWE 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.

Finally, let us consider the listing of the MWE-components: these must be listed in a way that is compatible with the syntactic structure assigned to the MWE. In the Rosetta system there are many highly theory- and implementation-specific aspects to this:

- the order of the MWE components is highly implementation-specific.
- the presence of certain components is highly theory or implementation specific: articles are never part of the MWE component list because articles are introduced syncategorematically by rules to form definite or indefinite noun phrases.

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 MWE 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 MWEs, that we have solved the problem

of uniquely identifying the MWE components, and their listing in the MWE 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. First, there is the possibility that the standard is not specific enough to allow for an automated procedure at all. But 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 MWEs should be described can be devised at all.

6 A Proposed Standard for MWEs!

Despite the conclusions drawn in the preceding section, I will propose a standard for MWEs in this section. The proposed standard covers fixed, semi-flexible and flexible MWEs. Moreover, it will be a proposal that avoids most if not all of the problems brought up, and — though this will require further work and research — the basic idea behind it lends itself to other types of MWEs (e.g. support verb constructions) as well.

The central idea behind the proposal is that the proposed standard does not prescribe the structure of an MWE, but backs off to a slightly weaker position, viz. it requires that it is specified which MWEs have the same structure. In short, it requires that equivalence classes of MWEs 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 MWE 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 MWEs will be used.

6.1 The proposed standard

In order to get concrete, I propose that an MWE description should consist of the following parts:

MWE pattern an identifier that uniquely identifies the structure of the MWE. The equivalence classes are defined with the help of these MWE patterns: MWEs with the same MWE pattern belong to the same equivalence class

List of MWE components This takes the form of a sequence of strings, each string representing the lexicon citation form[8] of each MWE component. The

[8] Usually the base form or lemma of the form. Of course, a common standard must be adopted here as well, but there are *de facto* standards for most languages.

list must contain a citation form for each MWE component (so articles, left out in the Rosetta approach, should be included). As to the 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.

Example sentence An example sentence that contains the MWE. An important additional requirement is that the structure of the sentence is identical for each example sentence within the same equivalence class. This can be achieved by using a fixed template for the example sentences.

Next to the MWE description, we need a description of the MWE patterns. This is a list of MWE pattern descriptions, where each MWE pattern description consists of two parts:

MWE pattern The MWE pattern identifier

Comments Free text, in which it is clarified why this MWE pattern is distinguished from others, further indications are given to avoid any possible ambiguities as to the nature of the MWE structure. It is even possible to supply a more or less formalized syntactic structure here (e.g. in the form of a tree), but the idea is that 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 MWEs. 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.

Example (11) shows 3 instances of the same MWE equivalence class from Dutch, and gives a description of the MWE pattern used to define this equivalence class

(11) a.

MWE pattern	MWE components	Example
MWEp1	de pijp uit gaan	Hij is de pijp uitgegaan
MWEp1	het schip in gaan	Hij is het schip ingegaan
MWEp1	de boot in gaan	Hij is de boot ingegaan

b. **MWEp1:** Verb taking a subject and a directional adpositional phrase (PP). This PP is headed by a postposition and has as its complement a noun phrase consisting of a determiner and a singular noun.

6.2 The conversion procedure

The procedure to convert a class of MWE descriptions made in accordance with the standard proposed, into a class of MWE 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 MWE 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 MWE pattern P consists of 5 steps:

1. Find an example sentence for MWE pattern P, and have it parsed by the system.
2. Use the structure resulting from the parse to determine the syntactic structure of the MWE in the system: define a transformation to turn the parse structure into the MWE structure.
3. Use the result of the parse to determine the unique identifiers of the lexical items used in the MWE.
4. Use the structure resulting from the parse to define a transformation to remove and/or reorder lexical items in the MWE component list.
5. Apply this transformation and make sure that the citation form of each lexical item equals the corresponding element on the transformed citation form list

The automatic part of the conversion procedure is applied to each instance of the equivalence class defined by MWE pattern P, and also consists of 5 steps:

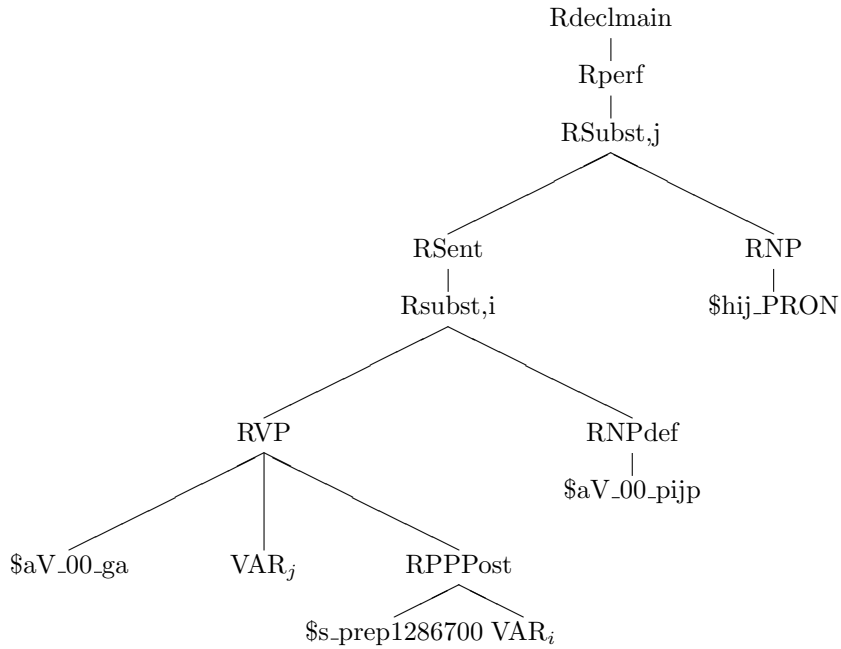
1. Parse the example sentence of the MWE and check that it is identical to the parse tree for the example sentence used in the manual step, except for the lexical items.
2. Use the transformation defined above to turn the parse tree into the structure of the MWE.
3. Select the unique identifiers of the lexical items' base forms from the parse tree, in order
4. Apply the MWE component transformation to the MWE component list
5. Check that the citation form of each lexical item equals the corresponding element on the transformed MWE component list

6.3 Illustration

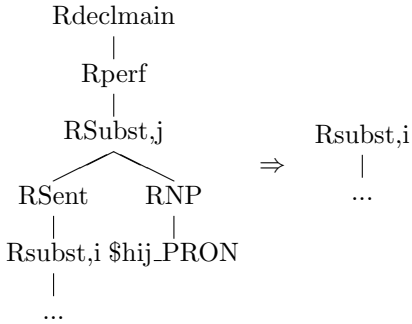
I will illustrate the procedure by applying it to the examples given above and deriving the representation required in the Rosetta system. We first apply the manual part for the MWE pattern MWEp1.

1. We select the example sentence *Hij is de pijp uitgegaan*. Parsing it by the Rosetta system yields the following syntactic D-tree:

(12)



2. We define a transformation to transform the parse tree into the structure of the MWE. The transformation can be defined informally as follows:



   i.e. delete everything above the node containing the (parameterized) rule *Rsubst,i*.
3. Given the resulting tree, the skeys of the lexical items of the idiom are
   - $aV_00_ga
   - $s_prep1286700
   - $aV_00_pijp

   in this order
4. The citation forms listed in the MWE component list (*de pijp uit gaan*) can be brought in correspondence with the skeys 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*.

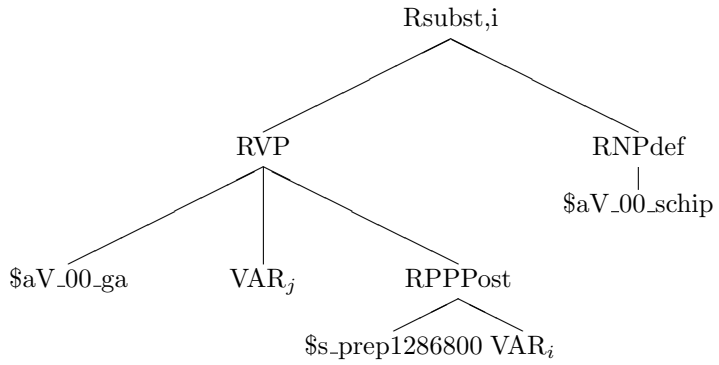
5. the citation forms of the skeys correspond to the elements on the transformed MWE component list:
   - citation form ($aV_00_ga) = *gaan*
   - citation form ($s_prep1286700) = *uit*
   - citation form ($aV_00_pijp) = *pijp*

In this way we have obtained a procedure to convert MWEs of MWE pattern MWEp1 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 MWE *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 one in (12), except for the skeys
2. The transformation turns it into the following D-tree:



3. The skeys for the lexical items in this D-tree are
   (a) $aV_00_ga
   (b) $s_prep1286800
   (c) $aV_00_schip
   in this order
4. The MWE component list transformation applied to the MWE component list *het schip in gaan* yields *gaan in schip*
5. The citation form of each lexical item equals the corresponding element on the transformed MWE component list:
   - citation form($aV_00_ga) = *gaan*
   - citation form($s_prep1286800) = *in*
   - citation form($aV_00_schip) = *schip*

Hence, we have derived the representation for the MWE *het schip ingaan* in Rosetta in a fully automatic manner:

- the syntactic D-tree (that can be replaced by a pattern identifier)

– the list of skeys used in the MWE, in an order compatible with the syntactic D-tree

If we want to represent the MWE literally as we showed above, with the lexical item for the head, and without the head in the MWE component list, we simply have to delete the first element of the MWE component list.

Some of the steps in the procedure may seem redundant. However, their role and use will become clear in the next section.

6.4 Possible problems and objections

In the preceding section we illustrated the procedure to derive a system-specific representation for MWEs 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.

We start with the manual part. The first step is to select an example sentence illustrating the MWE pattern, and to parse it. The resulting parse tree is then used in the further steps. If there is exactly one parse tree, the expert has to check whether this is the system-specific parse corresponding to the MWE pattern description. If it is, there is no problem, but what if it isn't? Or what if the example sentence does not yield a parse at all? This may certainly happen, but the fact that it can happen is in fact a virtue of the proposal: the proposed procedure does not allow one to include MWEs in one's system that one cannot parse correctly or at all. If it happens, one has to investigate the cause. These can be manyfold, e.g.

– the MWE contains words not contained in the system's lexicon
– all the MWE's words are contained in the system's lexicon, but one (or more) of them does not have the right properties to derive the relevant structure
– the MWE contains a construction not covered by the system's grammar

In all these cases, it is actually a virtue that one is pointed out that the system cannot handle this MWE: 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.

It is also possible (actually, probably the most frequent case) that the parse yields multiple structures. Again, it is possible that none of these multiple structures is the one corresponding to the MWE pattern. In this case, one will have to extend the lexicon and/or grammar again. Ignoring this case further, there are two possibilities:

1. the structure is ambiguous in aspects not directly related to the MWE, but only to the parts of the example sentence added. In that case an arbitrary selection can be made.

2. the ambiguity concerns the MWE part of the sentence. E.g., the Dutch MWE *de pijp aan Maarten geven* (lit. 'to give the pipe to Maarten', idiomatically 'to die', or 'to quit') will yield two syntactic structures in most systems: one with the phrase *aan Maarten* 'to Maarten' as a complement to the verb, and one with this phrase as a modifier to *pijp* 'pipe'. In this situation the free text comment describing the MWE pattern, and contrasting it with other MWE patterns should help the developer select the right parse. If it does not, the developer will have to rely on his/her linguistic and language knowledge to make a selection. If there are reasons to make a selection, the correct selection must be made. If there are none, an arbitrary selection can be made.

Of course, the fact that this kind of ambiguity can arise is a direct consequence of the fact that the standard does not supply the MWE structure, but only the equivalence classes of MWEs having the same structure. So, the proposed method requires effort not required in the stronger proposal that prescribes the structure of an MWE. However, as stated above, I doubt that the stronger proposal is feasible at all, and if it is feasible, whether it will require less effort than the weaker proposal pursued here.

It may also be the case that the developer, giving his/her knowledge of the system and the MWE pattern description, concludes that the current pattern collapses MWEs in a single equivalence class while his own system requires a further subdivision. In fact, the very examples we have used for illustration can probably not be treated uniformly in Rosetta: articles are introduced syncategorematically in the Rosetta system but other determiners are not. The pattern MWEp1, however, encompasses noun phrases in the postpositional phrase with all kinds of determiners.

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 arise in a proposal that prescribes how MWE structures look like–in fact it would arise much more often in such a proposal. The current proposal, however, is –in my view – as theory neutral as possible. I cannot imagine a proposal that yields at least the same positive results and is still more theory-neutral.

Even though a degree of theory-dependence is unavoidable, the proposed method still reduces effort even in cases where the theory-dependence shows up. For example, for the example cited, the problem is reduced to the examples of this equivalence class, and for these an automatic separation of the cases is easy to make (just search for examples containing articles as opposed to other determiners). Only a theory-dependence cutting through all or most equivalence classes, requiring rearrangements in all or most equivalence classes would require a severely more fine-grained subdivision on equivalence classes, and might make the current proposal less useful as a standard, and perhaps even undermine the very idea of a standard in this area at all.

We now turn to the automatic part of the procedure. In the first step, parsing the example sentence, there can be several different outcomes:

- No parse tree is produced at all.
- A single parse tree is produced, but it is not identical to the parse tree of the example sentence used in the manual part (ignoring the lexical items).
- More than one parse tree is produced

Again, the fact that these outcomes are automatically detected is a virtue of the approach. If no parse tree results, it would make no sense to add a lexical item for the MWE before the lexicon and/or grammar are adapted so that the syntax of the construction can be dealt with to begin with. If there is a parse tree, but one that is not identical to the manual part example sentence parse tree, it basically means that the grammar is not able to parse the sentence in the way it should be parsed, so again an adaptation of the lexicon and/or the grammar is called for. If more than one parse tree is produced, the sentence is considered ambiguous by the grammar, but the comparison with the manual part example sentence parse tree will automatically resolve this ambiguity. Of course, it is also possible that none of the multiple parse trees is identical to the manual part example sentence parse tree, but again this shows that the grammar is not able to parse the sentence in the way it should be parsed, and an adaptation of the grammar and the lexicon is called for.

In the second, third and fourth steps of the automatic procedure, nothing can go wrong if the classification in equivalence classes has been carried out correctly.

In the final step, it might be the case that one or more of the citation forms generated by the grammar from the ordered list of unique lexical entry identifiers do not correspond uniquely to the citation forms in the transformed MWE component list. Again, if this happens, it must be seen as a virtue of the approach, since it prevents one from using the wrong lexical items in the MWE expression. This situation can occur when one inflected word form is in fact a word form of two different lexical items, with different citation forms.[9] If such cases occur, the choice may often be arbitrary unless this MWE component allows for variations where the difference shows up.

Finally, there is one potentially real difficulty for the proposed method. 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. In addition, managing the various different equivalence classes will become more difficult, though this problem can be reduced by adopting an approach in which names of equivalence classes are constructed in a structured way. A method that prescribes the structure of each MWE, if feasible at all (which I doubt, as indicated above) would score better if this would be the case. One can expect that the number of different equivalence classes will be high, since in effect we translate structured objects (parse trees) into atomic objects (names of equivalence classes). In order to get a better sense of how high the number of equivalence classes can be and still be useful, it may be interesting to

[9] E.g. Dutch *sloten*, which can either be the plural of the noun *slot* 'lock', or the plural of the noun *sloot* 'ditch'.

note that in the Rosetta project we estimated the number of MWEs in Dutch on the basis of the electronic *Van Dale Hedendaags Nederlands* (contemporary Dutch) dictionary:[10] There are explicit markings in this dictionary for MWEs, and based on that we concluded that Dutch has approximately as many MWEs as it has single word lexical entries (90,000, in this case).[11] Assuming that this is a reasonable estimate, it would mean that even if the number of equivalence classes would be 9000, one can still achieve a reduction of effort by a factor 10 (only 9,000 MWE structures have to be considered instead of 90,000) when one reuses MWEs created independently following the standard proposed here.

7 Conclusions

In this paper I have analyzed the properties of three types of MWEs of increasing complexity (fixed, semi-flexible, and flexible MWEs) and how they are dealt with in two different machine translation systems. 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 MWEs. 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 MWEs in many systems that may differ widely in terms of their theoretical basis, their actual implementation, and their treatment of MWEs.

I have not discussed an important class of MWEs at all: support verb constructions and lexical collocations. However, I am convinced that the central idea behind the current proposal, viz. not describing what the syntactic structure of an MWE is but rather describing which MWEs have the same syntactic structure, 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.

We have seen one 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. Whether this is the case can only be found out by testing out the proposed method to a substantial number of MWEs.

References

[2000] Jan Odijk. Multilingual lexicons. ISLE Meeting, Paris, April 2000.

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

[10] Sterkenburg (1984).

[11] The estimate is tentative, since we found that the explicit markings for MWEs were not applied systematically.

[1986] André Schenk. Idioms in the Rosetta machine translation system. In *Proceedings of the 11th Conference on Computational Linguistics*, Bonn, 1986. Philips Research M.S. 13.508.

[1992] André Schenk. The syntactic behaviour of idioms. In Martin Everaert, Erik-Jan van der Linden, André Schenk, and Rob Schreuder, editors, *Proceedings of Idioms, International Conference on Idioms, Tilburg, The Netherlands, 2-4 september 1992*, pages 97–110. ITK , Tilburg University, 1992.

[1994] André Schenk. *Idioms and Collocations in Compositional Grammars.* PhD thesis, University of Utrecht, 1994.

[1984] P.G.J. Sterkenburg, editor. *Groot Woordenboek Hedendaags Nederlands.* Van Dale Lexicografie, Utrecht, 1984.