Chapter 1

Lexical encoding formats for multi-word expressions: The challenge of "irregular" regularities

Timm Lichte
University of Düsseldorf

Simon Petitjean
University of Düsseldorf

Agata Savary
University of Tours

Jakub Waszczuk
Université of Tours
University of Orléans

This chapter contributes a general overview and discussion of lexical encoding formats for multi-word expressions (MWEs) that can be used in NLP systems, in particular with large-scale grammars. The presentation is kept general in the sense that we will try to elicit basic aspects of lexical encoding and then elaborate on the specific sorts of challenges encountered when dealing with MWEs, especially the "irregular" regularities mentioned in the title. These insights will eventually be used to classify and evaluate different approaches to encoding. Even though this kind of evaluation cannot be conclusive given the diversity of languages and tastes, we will nevertheless argue in favor of fully flexible encoding formats exemplified with PATR-II and XMG, as opposed to the fixed encoding formats of DuELME and Walenty.

Timm Lichte, Simon Petitjean, Agata Savary & Jakub Waszczuk. 2019. Lexical encoding formats for multi-word expressions: The challenge of "irregular" regularities. In Yannick Parmentier & Jakub Waszczuk (eds.), *Representation and parsing of multi-word expressions: Current trends*, 1–33. Berlin: Language Science Press. DOI:10.5281/zenodo.2579033

1 Introduction

In this chapter, we seek to answer a seemingly simple question: what is it that makes an encoding format suitable for encoding multi-word expressions (MWEs) as part of an electronic resource? One quick answer could be: the encoding must be both machine- and human- readable, it must be factorized, and, last but not least, it must be able to cope with the specific irregularities of these objects. But what does this exactly mean? In fact, we claim that the casual use of "irregularity" actually threatens to cover a great deal of regularity, even though it is often a regularity that might look uncommon. In this chapter, we therefore aim to provide a more precise understanding of the underlying notions and concepts, and to apply this to a selection of formats which have a potential of encoding large classes of MWEs, including notably verbal ones, namely DuELME, Walenty, PATR-II and XMG. Thus, we are not aiming at the presentation of a comprehensive list of encoding formats ever proposed for MWEs, but rather want to elicit general aspects and typical examples thereof.

The chapter is structured as follows. We will first sort out general notions and principles of lexical encoding, starting with the notion of regularity in Section 2 and the notion of encoding in Section 3, and then turn to general virtues of lexical encoding formats in Section 4. Following this, in Section 5, we will go into more specific aspects, or rather challenges, that are to be dealt with when encoding MWEs. With this in view, we will then analyze existing formats by dividing them into two groups: fixed encoding formats will be treated in Section 6, and fully flexible ones in Section 7. In Section 8, we will finally compare the encoding formats and summarize the chapter.

2 On the notion of regularity

Regularity in the sense we are concerned with refers to the way properties are shared between the members of a set of objects. For now, we take a property to be just some atomic name and assume that every object is assigned exactly one subset of a given set of properties. We then say that a property p is REGULAR with respect to a set of objects E, iff p is shared by at least two members in E. Otherwise p is IRREGULAR (or IDIOSYNCRATIC). If p is regular but is shared only by a proper subset of E, we call p NON-TRIVIALLY REGULAR. By contrast, in the TRIVIALLY REGULAR case, p is regular and shared by all the objects in E. Here, p can be removed without harm because it does not distinguish any two objects in E. Sets of properties can be treated accordingly, hence a property set P is regular, if it is a subset of property sets of at least two objects in E. We then extend the notion of

regularity to objects by calling an object regular, if it only has regular properties and property sets, and otherwise irregular. Finally, this simplistic formalization allows for a straightforward characterization of the DEGREE OF REGULARITY, for example, in terms of likelihood (how likely is the property set of an object given a property distribution in the underlying object set) and diversity (how many property sets are found in an object set).

This notion of (ir)regularity implies that it is impossible to determine once and for all whether the properties of certain objects are regular or irregular, simply because the set of conceivable properties and objects is unbounded. In other words, the whole business of telling apart regularity from irregularity hinges on the selection of properties along with a specific set of objects.

Applying this to linguistics, the traditional view on the division of labor between syntax and lexicon is only valid for a specific set of linguistic objects, namely words, phrases and sentences, and a specific set of "syntactic" properties. Only on these premises is it valid to say that syntax is the realm of regularity whereas the lexicon is the collecting point for irregular aspects. To give an example, one could consider phrase structure rules as properties of words, phrases and sentences, depending on whether the phrase structure rules can be used to derive them. According to this set of properties, the words would be derived only by idiosyncratic rules that cannot be used to derive any other word. Hence, the set of words (= the lexicon) would not be fully regular, other than the sets of phrases and sentences (= the syntax). However, when taking other properties into account such as semantic, morphological and phonological ones, this division becomes blurred quite easily.

Similarly, if an MWE (or some property of it) is called "irregular", this can have at least one of three possible reasons: (i) the set of objects is sufficiently restricted (e.g., by contrasting the MWE with non-MWEs only), or (ii) the set of properties is sufficiently extended (e.g., by taking into account very specific properties of the MWE), or (iii) the property set of the MWE is relatively unlikely and "irregular" is assigned a likelihood related meaning. In all three cases, there is actually a high risk of overlooking or neglecting some regularities, even more since we are dealing with objects that have not been in the center of interest in most of the mainstream grammar theories. This gives a hint of how we want "irregular regularities" from the title to be understood: as regularities that concern unusual properties. The assumption throughout this chapter will be that the irregularity of MWEs can be attributed to very few properties concerning the syntax-semantics interface, while there is a great deal of non-trivially regular properties that are shared across MWEs and permeate all levels of linguistic descriptions.

3 The most basic encoding format

Given what has been said in the last section, it should be fairly easy to see that the most basic encoding format of the properties of an MWE is via PROPERTY NAME SETS. Two examples for *kick the bucket* and *spill beans* are shown in (1):

(1) a. kick-the-bucket := {NP_0 V NP_1, NP_1.Det.the, NP_1.N.bucket, V.kick, meaning=die}
b. spill-beans := {NP_0 V NP_1, NP_1.N.beans, V.spill, passive, meaning=divulge}

Even if the property names seem to have some compositional structure (NP_1.Det.the means that the determiner of the object NP is *the*), they are chosen here for purely mnemonic reasons – one could have equally written something alphabetically innocent like p_{23}. So, in order to proceed, what is needed is an INTERPRETATION FUNCTION from property names to objects of whatever target formalism is chosen. Essentially, this is the characteristic of any encoding format, even the more sophisticated ones. Of course, there is some variance as to how close the encoding format is related to the target formalism. Daelemans & van der Linden (1992) refer to this aspect as notational adequacy. But be aware that, in our view, the adequacy of a lexical encoding format is multi-aspectual (see Figure 1 on page 6) and ultimately *user-oriented*. We will elaborate more on this in Section 4.

Speaking of the adequacy of property name sets, there are, in fact, some attractive properties of this very simple way of encoding: (i) it is very flexible in terms of adding and removing property names and adapting the interpretation function to some target formalism; (ii) it makes empirically largely neutral descriptions available; (iii) it is conceptually lean and inviting for formal novices because the main data structures are just ordinary sets. On the other hand, it is obvious that nobody would seriously make use of property name sets when encoding a large electronic lexicon – at least not without a tool that helps to ensure correctness by accounting for, and therefore encoding underlying generalizations, that is, patterns of co-occurrence among properties. Furthermore, one would need tools to specify and carry out the interpretation function. In our view, this does not only hold for pure property name sets; the actual encoding format is *always* surrounded by tools mediating towards the human user, the target formalism or the electronic resource – to what degree depends on the encoding format in question (see Section 4).

A closely related but more transparent encoding format is based on tables in which the rows correspond to lexical entries, or any other sort of object, and

Table 1: Table encoding of the property name sets in (1)

ID	NP_0 V NP_1	NP_1.det	NP_1.N	V	passive	meaning
kick-the-bucket	+	the	bucket	kick	–	die
spill-beans	+		bean	spill	+	divulge

the columns to properties. Binary cell values then indicate whether a property holds for an object or not. This format has gained some popularity, for example, through the extensive work of Maurice Gross (and colleagues) within his lexicon-grammar framework (Gross 1994). While lexicon-grammar matrices are binary, at least for the most part, a larger range of cell values helps to yield a more succinct matrix. This is shown in Table 1 which translates the property sets from (1). Needless to say, for any such non-binary matrix, there is an equivalent binary one with a larger number of columns or properties.

The table format makes the presentation of property name sets more readable, but apart from this, it comes with very similar methodological implications: it is suitable for collecting observations, but it cannot express recurring patterns within these observations, that is, a theory. For this, and thus also for ensuring correctness and completeness, additional tools are needed.

4 General virtues of lexical encoding formats

The preceding section showed that certain encoding formats stand out in terms of simplicity and accessibility, but also manifest critical drawbacks as to usability and expressivity. This section tries to sort out more systematically the diverse and sometimes contradicting virtues an encoding format can have. The cause of diversity is not hard to pinpoint: it is the interface status of encoding formats, as illustrated in Figure 1, with similarly diverse conjugates, namely a human user, a lexical object and a lexical resource.

4.1 Encoding virtues with respect to a lexical object

We already learned in Sections 2 and 3 that the simplest conception of a lexical object and an encoding format is a set of properties or property names. Let P_i be the property set of a lexical object. An encoding of P_i is a property name set P_i^e together with an encoding function which maps P_i onto P_i^e. Hence, the encoding examples given in (1) on page 4 are actually accompanied by an imagined lexical

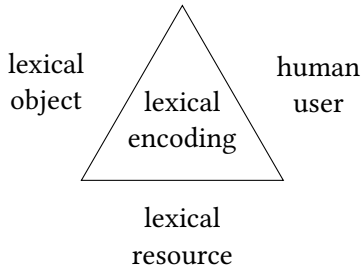

Figure 1: Interface aspects of lexical encoding

object and an encoding function. It is furthermore important to keep in mind that, for now, we ignore inferential means of encoding formats that help to express generalizations, that is, we assume that encodings are fully resolved.

Based on this understanding of encoding, the encoding virtues are easy to see and capture, namely, the encoding of a property set P_i should be complete and concise. An encoding (function) is COMPLETE *iff* every property of P_i is mapped onto a property name of P_i^e. Thus the encoding function is injective. On the other hand, an encoding is CONCISE *iff* for every encoding property p_i^e there is a source property p_i such that p_i^e is the encoding of p_i. Here, the encoding is surjective. In other words, no property name is added unmotivatedly. Of course, an encoding should be both complete and concise, and consequently the encoding function should be bijective. This implies that distinctions made in P_i are minimally preserved in the encoding of P_i.

To give an example, Table 1 is a complete encoding of the property sets in (1). Yet it is not perfectly concise: the property set of *kick-the-bucket* does not have a passive feature, while there is a passive cell in the table encoding. Similarly, the $\text{NP}_\mathbf{1}$.det cell in the encoding of *spill-beans* does not have a corresponding property in the source set. Still, the encoding in Table 1 appears to be only slightly less concise than the original property sets in (1), and moreover the table encoding is (in most cases) more accessible for the human eye. This teaches us two things: (i) the validity of some encoding virtues can be a matter of degree, and (ii) they may conflict with other encoding virtues.

But before turning to possibly conflicting encoding virtues having to do with other aspects of encoding, let us finally have a look at the encoding of *sets* of lexical objects. Here, it is clearly desirable for an encoding to be CONSISTENT, simply meaning that the relation between the properties appearing in all the lexical objects under consideration and the target properties of the encoding is functional as well. This clearly holds for the encoding in Table 1 where identical properties are encoded as identical cell values within the same row.

4.2 Encoding virtues with respect to a human user

When it comes to the human user, a lexical encoding should be transparent, flexible and sufficiently powerful to capture generalizations.

By TRANSPARENT we mean that the human user should be able to map the encoding back to the source set of lexical properties. Needless to say, the degree of transparency very much depends on the taste and reading habits of the user in question. It could well be, although it is rather unlikely, that some users will feel more comfortable with plain property sets also when dealing with larger lexicons. Depending on the degree of training, it is even imaginable that users become fluent in rather opaque encoding languages that make use of property names such as p_{23}. This is, of course, not what we consider desirable: lexical encodings should not come with notational idiosyncrasies that keep novices away or are prone to lead to encoding errors (e.g., by misremembering p_{23}). Thus, since we are dealing with computational lexicons, we conceive an encoding language as transparent *iff* it is (i) mnemonic as to the property names and their denoted properties and (ii) precise by means of a rigorous denotational semantics to avoid vagueness and thus inconsistencies.

Since transparency is so important to the human user, but at the same time human users and also lexical objects can differ to a great deal, another crucial virtue of encoding formats is FLEXIBILITY. Lexical encoding usually is an incremental process where unforeseen properties can be encountered or the denotation of a property may change over time. A flexible encoding format allows the user to freely choose property names and to include new properties on the fly.[1]

Closely related to flexibility is the POWER TO GENERALIZE. With an increasing number of lexical objects that are encoded in a lexicon, usually also the desire to factorize the property sets increases in order to avoid redundancy. In other words, one would like to group properties and assign them collectively. Again, the human encoder should be free to choose the content and name of property subsets, or, more technically speaking, the parts of encodings should be reusable at any level of representation and detail. What may sound like a nice add-on is in fact a necessary prerequisite to express any non-trivial lexical generalization, such as that a passive construction does not include an accusative object.

Finally, we can consider an encoding format to be IMPLEMENTATION-FRIENDLY *iff* there exist tools that assist a human user with encoding large sets of lexical objects, or with verifying these encodings. This virtue already touches upon one aspect that will be also dealt with in the next section, which is the existence of software tools that help to convert lexical encodings into a lexical resource.

[1] Of course, flexibility also helps to keep the encoding complete in the sense of Section 4.1.

4.3 Encoding virtues with respect to a lexical resource

A lexical resource is an electronic representation of lexical encodings that can be (more or less) directly used in NLP applications. Accordingly, the virtue of ELECTRONIC VERSATILITY assigned to lexical encoding formats describes the relative ease with which a corresponding lexical encoding can be converted into a lexical resource. This easiness can allude to at least two different aspects; either the properties of existing conversion tools or the engineering task to produce them. Ultimately, what really matters when mapping a lexical encoding to an electronic resource is the mere existence of software tools to achieve this. Obviously, this is not a property of the encoding format itself, but a property of its interface with the specific format of an intended lexical resource. Thus, in this view, an encoding format would be electronically versatile whenever there exist many (and among them the desired) conversion tools. From the perspective of the programmer, however, electronic versatility has a different implication: it is rather related to the efforts it takes to implement such a conversion tool from scratch.

Even worse, it's certainly hard to say something conclusive about electronic versatility in global terms, as there is no true one-to-one relation. NLP applications can vary distinctively in their interface specifications, and therefore there is rather a one-to-many relation between a particular lexical encoding and the lexical resources that one might wish to derive from it. In the simplest case, the lexical encoding can act as the lexical resource proper. Yet, presumably in the majority of cases, the lexical encoding will be preprocessed and converted into something *less* user-friendly. This is most obvious in graphically enhanced encoding methods where the lexical resource is derived from the underlying, non-graphical representation. But, of course, this also holds for interchange formats such as LMF (Francopoulo et al. 2006), which are meant to provide a mediating standard and rely on cumbersome XML or the like.

Another relevant property of the interface between the lexical encoding and the lexical resource seems to be whether the generalizations expressed in the lexical encoding are preserved during conversion, or whether only fully resolved entries are included. From the point of view of the encoder, the availability of generalizations seems to be preferred, but this is a virtue of the lexical resource proper, and also depends on the targeted NLP application.

Summing up, electronic versatility is an important but also complex virtue that covers orthogonal, or even conflicting, aspects of the interface between lexical encodings and lexical resources. Moreover, given the heterogeneity of the latter ones, a general verdict is often difficult to obtain.

5 Challenges posed by MWEs

From a general point of view, MWEs are in no way different from any other lexical object: they can be encoded using property name sets as in (1) or using the table format from Table 1. But what is then so challenging about MWEs? On the one hand, it is the peculiarity of the affected properties, for example, the property NP_1.*Det.the* in the property set of *kick the bucket*. This is challenging with respect to the flexibility of an encoding format. On the other hand, the interactions between these and other properties pose a challenge to the power of an encoding format to generalize. In this section, we will go through some of these challenging properties and interactions, confining ourselves mainly to syntax and morphology.

Let us first examine a multilingual set of MWE examples[2] together with their peculiarities, which the MWE-related literature often calls irregularities or idiosyncrasies. In what follows, each property is either DEFECTIVE or RESTRICTIVE. In the former case, it excludes a literal interpretation of a given object. In the latter, it reduces the number of possible surface realizations of a given object with respect to the corresponding literal interpretation.

1. defective agreement, e.g. in (FR) *grands-mères* 'grandmothers' the adjective does not agree with the noun in gender, unlike most regular adjectival modifiers;

2. restrictive agreement, e.g. (EN) *to cross one's fingers* imposes agreement in person, number and gender between the possessive pronoun and the subject: #*I cross his fingers*

3. restrictive paradigm, e.g. (PL) *zjadłbym konia z kopytami* (lit. *I would eat a horse with its hooves*) 'I am very hungry' can only occur in conditional mood: #*zjem konia z kopytami* 'I will eat a horse with its hooves';

4. defective subcategorization, i.e. imposing a subcategorization frame which the MWE headword does not admit outside MWEs, e.g. (PL) *dobrze mu z oczy patrzy* (lit. *well him looks from eyes*) 'he looks like a good person' prohibits a subject: **uczciwość dobrze mu z oczy patrzy* (lit. *honesty well him looks from eyes*), while *patrzy* 'looks' as a standalone verb always requires one;

[2]Each example is preceded by its language code in parentheses. The hash (#) character signals the loss of the idiomatic reading due to a missing property, while the asterisk (*) means ungrammaticality.

5. restrictive diathesis, e.g. (EN) *to kick the bucket* does not allow passivization: *#the bucket was kicked*, while (FR) *les carottes sont cuites* (lit. *the carrots are cooked*) 'the situation is hopeless' only allows passive voice: *#on cuit les carottes* (lit. *one cooks the carrots*) ';'

6. restrictive choice of determiners and modifiers, e.g. (FR) *avoir raison* (lit. *to have reason*) 'to be right' allows neither a determiner nor a modifier of the nominal component: *#avoir (une) raison évidente* 'to have an obvious reason';

7. restrictive dependencies between determiners and modifiers: (FR) *avoir envie* (lit. *to have desire*) 'to feel like' admits no determiner for the predicative noun *envie* 'desire', if it takes no argument or modifier, or if it takes an infinitival argument governed by the preposition *de* 'of': *j'ai envie de le faire* (lit. *I have desire of to do it*) 'I feel like doing it'; but if the noun is modified by an adjective, the determiner is compulsory: *j'ai une envie folle de le faire* (lit. *I have a crazy desire of to do it*) 'I feel a lot like doing it';

8. restrictive modification, e.g. (FR) *mener une vie (de riche)* 'to live a life (of a rich)' imposes an adjectival or a prepositional modifier on the nominal: *#il mène une vie* 'he leads a life';

9. restrictive linearization, e.g. (EN) *drink and drive* requires the strict order of its coordinated verbs, violating this constraint leads to the loss of the idiomatic reading: *#drive and drink*;

10. restrictive lexical selection, i.e. imposing particular lexical realizations of certain syntactic arguments, e.g. (EN) *to pull someone's leg* requires the head verb *pull* with a direct object headed by *leg*: *#to pull one's arm/member*.

Note that while the above properties are perceived as unexpected or unpredictable, they are most often shared with other MWEs, therefore, in our understanding (cf. Section 2), they are regular. To make this more precise, recall that regularity of a property is not absolute but relative to a given set of objects E. In linguistic modeling, we tend to group objects into sets based on their similarities rather than their discrepancies. For instance, in valence-oriented modeling (such as Walenty or PART-II described in Sections 6.2 and 7.1, respectively, or IDION and the MWE lexicon of NorGram discussed in Markantonatou et al. (2019 [this volume]) and Dyvik et al. (2019 [this volume]) respectively), verbal constructions are grouped according to the lemma of their head verb, whereas in more

constructionist approaches (like DUELME and XMG, introduced in Sections 6.1 and 7.2), they are grouped by the syntactic structure of their subcategorization frames. Such properties used to group objects become trivially regular properties of these groups (since they are shared by all objects of a group). Most other properties have a varying degree of regularity and are only rarely truly idiosyncratic.

As an example, let us consider a set of English verbal expressions, each of which is headed by a verb, taking a subject and a direct object, and admitting modifiers, e.g. (EN) *John pulled the heavy door*. In this set, the property of allowing any head verb with the proper subcategorization frame is much more regular than restricting it to the verb *kick*. Furthermore, the property of allowing passivization is more regular than prohibiting passive voice, like in *John kicked the bucket* 'John died'. Also, allowing a possessive determiner of the object, as in *John pushed the/my door* is more regular than imposing it, as in *John broke his/her/our fall* 'John made his/her/our fall less forceful', which itself is more regular than imposing a possessive which agrees with the subject, as in *John crossed his fingers*. This last property is, however, still regular. In order for it to be idiosyncratic, *John crossed his fingers* 'John wished luck' and *John held his tongue* 'John refrained from expressing his view' could not co-occur in the same object set, which would hinder the usability of such a set for linguistic modeling. Without resorting to such artificial choice of object sets, Property 10 is one of the rare truly idiosyncratic properties, since it is usually specific to one MWE only, except in case of truly ambiguous MWEs like *to go on* 'to continue, to happen'.

Note finally that one MWE usually exhibits different properties of varying degrees of regularity. For instance, while the components of (FR) *grands-mères* 'grandmothers' do not agree in gender, they do agree in number. While (PL) *zjadłbym konia z kopytami* (lit. *I would eat a horse with its hooves*) 'I am very hungry' requires conditional mood, it has a highly regular inflection for person and number. While the object in (EN) *to pull someone's leg* is partly lexicalized, the subject is not. While (EN) *to kick the bucket* cannot be passivized, it does admit a restricted number of internal modifiers as in *to kick the proverbial bucket*, etc.

As a conclusion, the challenging nature of MWE is manyfold: (i) regularity of properties of MWEs is scale-wise, (ii) properties of different degrees of regularity co-occur in each MWE, (iii) truly idiosyncratic properties are rare (under the usual similarity-oriented grouping strategies), (iv) shared properties can be unforeseen (cf. Property 7), so listing them all in advance is hard. A general-purpose encoding format should possibly face all these challenges simultaneously. Note also the similarity of observations (i) and (ii) with the notion of a *flexibility continuum* in idioms, discussed in Sheinfux et al. (2019 [this volume]).

6 Fixed MWE encoding formats

While lexical approaches dedicated to a large variety of MWEs have a relatively long linguistic tradition, notably with Gross (1986) and Mel'čuk et al. (1988), NLP-oriented work on lexical encoding of MWEs has mainly dealt with continuous instances (Savary 2008). More recently, proposals have been put forward which also take verbal MWEs into account whose components are discontinuously linearized. Here, we study two instances of such approaches tailored to specific languages: DuELME (Grégoire 2010) for Dutch and Walenty (Przepiórkowski et al. 2014) for Polish. They stand out as: (i) having been designed with a (relative) theory-neutrality in mind, (ii) having resulted in MWE lexicons of several thousands of entries, (iii) having been coupled with real-size grammars, so as to test their usability for parsing. At the same time, DueLME and Walenty can be characterized as fixed encoding formats in the sense that their encoding language (basically the set of property names and their interpretation) cannot be freely chosen or extended.

6.1 DuELME

DuELME (Dutch Electronic Lexicon of Multiword Expressions, Grégoire 2010) is an electronic lexicon comprising roughly 5,000 Dutch multiword expressions.[3] It distinguishes two sorts of descriptions, pattern descriptions and MWE descriptions, which are composed of non-intersecting sets of predefined fields. Patterns, also called *parameterized equivalence classes*, represent mainly the syntactic structures of MWEs and the part-of-speech tags of their leaves. MWE descriptions express MWE-specific lexical and morpho-syntactic constraints.

Figure 2 shows a sample pattern (Lines 1–5), called `ec1`, and a MWE entry (Lines 7–11) assigned to it: (NL) *zijn kansen waarnemen* (lit. *one's chances perceive*) 'to seize the opportunity'.

The pattern describes expressions headed by a verb, taking a direct object consisting of a fixed determiner and a modifiable noun. The POS-entitled Line 3 lists the parts of speech of MWE components. The PATTERN-entitled Line 4 shows the syntactic structure, roughly, as a dependency tree where syntactic categories (`VP`, `NP`, `D`, `N1`[4], `V`) and dependency labels (`obj1`, `det`, `hd`) are marked explicitly, and some of the leaves are indexed (`1`, `2`, `3`) so as to be matched with components of a

[3] http://duelme.clarin.inl.nl/

[4] The `N1` category denotes an NP of which some elements are lexically fixed, but which is still subject to standard grammar rules such as agreement

```
1  % Pattern description
2  PATTERN_NAME ec1
3  POS d n v
4  PATTERN  [.VP [.obj1:NP [.det:D (1) ] [.hd:N1 (2) ]] [.hd:V (3) ]]
5  DESCRIPTION Expressions headed by a verb, taking a direct object
       consisting of a fixed determiner and a modifiable noun.
6
7  % MWE description
8  EXPRESSION zijn kansen waarnemen
9  CL zijn kans[pl] waar_nemen[part]
10 PATTERN_NAME ec1
11 EXAMPLE hij heeft zijn kansen waargenomen
```

Figure 2: DuELME pattern description ec1 (from Grégoire 2007b) and MWE description of (NL) *zijn kansen waarnemen* (lit. *one's chances perceive*) 'to seize the opportunity' (from Grégoire 2010)

particular MWE. Thus, the components *zijn* 'one's', *kansen* 'chances' and *waarnemen* 'perceive' of the MWE in Lines 8–9 are implicitly co-indexed with the `det:D`, `hd:N1` and `hd:V` nodes in the `ec1` pattern. Moreover, the component list (CL) in Line 9 lists the MWE-specific values of the "parameters" for the pattern, i.e. the lemmas of all components, as well as some morphosyntactic constraints, here: *kans* 'chance' must be in plural (`pl`), and *waarnemen* 'perceive' is a separable particle verb (`part`).

This approach is constructionist in the sense that MWEs are grouped into sets based on their structure (rather than their headword). While the syntax of patterns seems theory-specific, they might be seen rather as identifiers of equivalence classes, allowing to group MWEs of the same structure, whatever the syntactic formalism used to express this structure.[5] DuELME's view of the regularity is binary, which is reflected by its two-level description paradigm. Namely, it is assumed that each type of a syntactic structure has some "generally regular" properties covered by general grammar rules. These properties are not described in the lexicon but symbolized by patterns. Conversely, the MWE-specific properties are described in MWE entries. For instance, while the number of *kans* 'chance' is restricted to plural in Line 9, its other grammatical features are not specified since they are supposedly governed by grammar rules. This principle avoids some grammar vs. lexicon redundancy. Note, however, that the choice of properties to be included in patterns is rather arbitrary and in most cases leads

[5]Jan Odijk, personal communication 21 September 2015.Odijk, Jan@Odijk, Jan

to partly redundant descriptions. For instance, the `part` property in Line 9 is shared with other MWEs containing separable particle verbs, and has to be specified for each of them. This redundancy at the level of MWE descriptions could be avoided, if the `ec1` pattern were restricted to `d-n-v` constructions containing separable particle verbs only. This would, however, require a new pattern with the same structure but a different verb type selection, in order to cover e.g. (NL) *zijn debuut maken* (lit. *to make one's debut*), which would lead to redundancy at the level of patterns. Since there is no notion of reference, or reuse, among the 141 pattern descriptions that DuELME comprises (Grégoire 2007b), such redundancy could not be avoided.

As a conclusion, the distinction between patterns and MWE descriptions introduces a limited degree of factorization. While some syntactic constraints, e.g. dependencies, are mentioned more or less explicitly in patterns, some other syntactic properties are implicit (supposed to be covered by the grammar and known to the NLP system). Some specific constraints, e.g. restrictive agreement, diathesis, determination, modification and linearization, discussed in Points 2 and 5–9 in Section 5, seem not possible to express. The interpretation of the encoding is led partly by the syntax of patterns and entries, and partly by textual documentation (Grégoire 2007a), where it is sometimes hard to distinguish formal properties and inference rules from methodological strategies and recommendations, i.e. the transparency level of the format is relatively low. Lastly, the format is not flexible, i.e. extending the set of describable properties can only be done ad hoc rather than within an established framework with a clear denotational semantic.

It is worth noting that DuELME benefits from a standard LMF format (Odijk 2013), which makes it more electronically versatile, even if it does not seem implementation friendly in the sense that tools supporting lexicographic encoding in this format do not seem publicly available.

6.2 Walenty

A quite different encoding style is found in Walenty, a Polish large-scale valence dictionary that includes an elaborate phraseological component (Przepiórkowski et al. 2014; 2016). It contains over 100,000 syntactic frames, 14,000 of which are verbal frames with lexicalized arguments, i.e. verbal MWEs. An entry in Walenty contains a headword (here a verb), followed by a list of argument descriptions (separated by +).

Figure 3 shows a (slightly simplified) sample MWE entry of (PL) *dobrze [KOMUŚ] z oczu patrzy* (lit. *well someone.*DAT *from eyes looks*) 'someone looks like a

```
1 patrzeć: np(dat)+advp(misc)+lex(prepnp(z,gen),pl,'oko',natr)
```

Figure 3: Description of *dobrze [KOMUŚ] z oczu patrzy* (lit. *well someone.DAT from eyes looks*) 'someone looks like a good person' in Walenty

good person', which exhibits several interesting constraints. Firstly, the syntactic subject is prohibited here, which is expressed simply by omitting the `subj` argument in the valence frame. Secondly, the indirect object in dative is compulsory (`np(dat)`). These two properties are unusual, since *patrzeć* 'look', as a stand-alone verb, does take a subject and it only admits an indirect object with prepositional complements headed by *na* 'on' and *w* 'in'. Thirdly, the adverb *dobrze* 'well' can have some variations, e.g. *źle [KOMUŚ] z oczu patrzy* (lit. *evilly someone.DAT from eyes looks*) 'someone looks like an evil person', therefore it is encoded by a more generic, non lexicalized, `advp(misc)` requirement of a "true" adverbial clause.[6] Finally, within the lexicalized prepositional group (`lex(prepnp(…))`, which does not admit modification (`natr`), the preposition *z* 'from' governing the genitive case (`(z,gen)`) requires its nominal complement to be a plural form of the lemma *oko* 'eye' (`pl,'oko'`).

This approach is valence-based, i.e. MWEs are seen as particular syntactic frames of their head verbs, in which some arguments happen to be (at least partly) lexicalized. Regularity is implicit: "generally regular" properties are supposed to be covered by grammar rules and only MWE-specific properties are expressed in lexicon entries. E.g., while the plural number of *oko* 'eye' is specified, its case is not, since it is supposed to regularly agree with its governing preposition (which requires genitive case). This principle is similar to the one admitted in DuELME (cf. Section 6.1), here however, no equivalence classes are used, so the syntactic structure, understood as the list of arguments (possibly structured themselves) required by the head verb, is encoded in each entry (similarly to the IDON lexicon discussed in Markantonatou et al. (2019 [this volume])), which leads to redundancy in the lexicon. For instance, entries for all MWEs taking a non-lexicalized subject, direct object and indirect object, and a partly lexicalized prepositional complement, contain the same sequence: `subj{np(str)} + obj{np(str)} + {np(inst)} + {lex(prepnp(…)}`[7]. Some redundancy can, however, be avoided due to macros which encode some repetitive substructures. For

[6] A "true" adverbial clause cannot be realized by a prepositional nominal group.

[7] The `str` feature stands for a structural case. For the subject, it is usually nominative, but it turns to genitive when the expression is nominalized. For the direct object, it is accusative but it turns to genitive when it occurs under the scope of negation.

instance, the `possp` macro encodes all possible realization of a possessive phrase, including nominal phrases with genitive and possessive determiners like *mój, czyjś, własny, …* 'my, one's, one's own, …'.

Some additional syntactic properties can be expressed on the level of the whole MWE, e.g. the fact that the head verb is perfective or imperfective, that the MWE must always contain negation, or that it can or cannot be passivized. Some other types of constraints, e.g. restrictive agreement, paradigm, determination, or linearization (cf. Points 2–3, 6–7 and 9 in Section 5), exceed Walenty's expressive power. Therefore, one cannot express the fact that, in (PL) *dobrze [KOMUŚ] z oczu patrzy* (lit. *well someone.*DAT *from eyes looks*) 'someone looks like a good person', the head verb *patrzeć* 'look' is always in the 3rd person singular (any tense or mood), although it has a complete inflection paradigm as a stand-alone verb.[8] Also, there is no means to specify that the adverb *dobrze* 'well' should usually precede the prepositional complement and the verb.[9] Note, however, that a conservative extension of the formalism to include some of these constraints was proposed by Przepiórkowski et al. (2016).

The interpretation of the encoding is led partly by the syntax of entries and explicit macro extensions, and partly by the accompanying textual documentation. Some inferences remain unclear, e.g., some macros contain non-documented shortcuts, and some codes have no clear denotational semantics. The format is rather inflexible, that is, extending the set of describable properties can only be done ad hoc. Walenty does benefit from a standard interchange XML metaformat, namely TEI[10], but does not provide its precise instantiation in terms of a DTD, RelaxNG or XML schema. Finally, it has a rather elaborate lexicographical support, with several user roles, where the existing entries can be browsed together with their corpus examples, and new entries can be added, corrected, compared, assigned to users, etc. (Nitoń et al. 2016). Recent developments couple Walenty with a Polish wordnet so as to enrich valency data with semantic frames.

7 Fully flexible encoding formats

What we mean by fully flexible is that properties, property names and inference rules (or macros) can be freely chosen – one consequence being that there are

[8] Impersonal (i.e. allowing no subject) finite verbs typically occur in the 3rd person singular in Polish, so the expression of this fact is probably left to the grammar. If so, then this fact seems implicit.

[9] A different word order would be considered as marked.

[10] Text Encoding Initiative: http://www.tei-c.org/Guidelines/P5/

usually many ways to implement an object within such an encoding format. In this section, we will show two exemplars of fully flexible encoding formats: the venerable PATR-II and the more recent XMG. The motivation for choosing these two encoding formats is twofold. On the one hand, both engage different notational means with a different denotational semantics; on the other hand, two extremes of modeling argument structure can be covered that were the focus of some debate recently, namely the lexical versus the phrasal approach (Müller & Wechsler 2014). In doing so, we will again, as in the preceding section, restrict ourselves to the tentative encoding of (NL) *zijn kansen waarnemen* 'to seize the opportunity' and (PL) *dobrze [KOMUŚ] z oczu patrzy* 'someone looks like a good person'. The presentation will, we think, strengthen the view that MWEs should be better encoded with fully flexible encoding formats in order to obtain and maintain the virtues mentioned in Section 4.

7.1 PATR-II

A true classic, PATR-II (Shieber 1984; 1986) dates back to the early 80s and has greatly influenced the development of later encoding formats, for example LKB (Copestake 2002: 6), thanks to its notational transparency and conceptual rigor.[11] The basic idea is simple: to enhance CFG rules with descriptions of untyped feature structures, which are then unified during rule applications. Hence, the models of PATR-II descriptions are just directed acyclic graphs with labeled nodes and edges. But the means of description are more elaborate and do also include templates, lexical rules and sometimes – depending on the PATR-II implementation – default inheritance.[12] The encoding examples that we will give do not, however, make use of the full non-monotonic power of PATR-II, as lexical rules and default inheritance will be left out. On the other hand, we will follow the head-driven perspective of PATR-II in that MWEs will be encoded in their head only, that is, MWEs headed by a verb will essentially emerge from the encoding of their verbal component.[13]

[11] A superficially similar encoding framework is DATR (Evans & Gazdar 1996). See Kilbury et al. (1991) for a comparison with PATR-II that also highlights the considerable differences between the two.

[12] Default inheritance is available, for example, in PC-PATR (McConnel 1997), which is a parser for PATR-II grammars developed at the Summer Institute of Linguistics (SIL).

[13] The only previous work on encoding MWEs with PATR-II that we are aware of is found in Habert & Jaquemin (1995). There, the focus is on French nominal compunds like *verre à vin* ('wineglass').

All this is exemplified for (NL) *zijn kansen waarnemen* in Figure 4. Templates are headed by `Define-as` constructs. The body of a template may either contain template names (or disjunctions thereof as in Line 33), from which the template inherits, or feature structure descriptions. Word entries such as the one of *waarnemen* at the bottom are similiar to templates but define the terminals of CFG rules. Keep in mind that *waarnemen* acts as the verbal head of the MWE, hence the templates in this example all describe the feature structure of *waarnemen* only. Also note that the features are chosen to keep the example as simple as possible – typically one would find subcategorization lists in PATR-II implementations.

In Figure 4, the first five templates (`Verb`, `Subject`, `Object`, `Intransitive`, and `Transitive`) just act as an example of how general properties, like being a transitive verb, *could* be factorized into even more general properties. Finally, the sixth template, `SubjectPossObjectAgreement`, is more immediately relevant to the MWE (NL) *zijn kansen waarnemen* since it captures the agreement of the subject with the possessive pronoun at the object. This is achieved by using the shared variable `$1`. Crucially, this template could be reused in many other MWEs such as (EN) *to do one's best*. Again, this is not to say that this sort of agreement should be treated in this way, but that it is *possible* to do so, choosing here just one of the many available options. In other words, the template `SubjectPossObjectAgreement` is an instance of one of such MWE-specific regularity that PATR-II is flexible enough to encode directly. Finally, in Figure 4, the template `ZijnKansenWaarnemen` inherits from the templates `Transitive` and `SubjectPossObjectAgreement`, and it adds further information on the shape and modifiability of the object and on the idiomatic semantics of the whole MWE.

Comparing the PATR-II encoding with the DuELME encoding from Figure 2, it becomes evident that PATR-II is more flexible at defining properties or factorizing what are called "patterns" in DuELME. The reason for this divergence of flexibility also lies in the fact that PATR-II descriptions come with a clear denotational semantics, which does not seem to be fixed for DuELME encodings. In fact, one could see this as an advantage of DuELME, taking it as a sign of desired neutrality. But then one must also accept intransparency and inflexibility, at least to some degree.

A tentative PATR-II encoding of (PL) *dobrze [KOMUŚ] z oczu patrzy* is presented in Figure 5. As explained in Section 5, the challenge with this MWE is a mixture of particular constraints regarding the subcategorization frame of the verb (*patrzy* 'looks' is used as an impersonal transitive) and the sentence initial linearization of the adverb. The encoding example in Figure 5 takes care of this by stipulating special features that would trigger the right CFG rules at the right

```
Define Verb as
    [cat: v]

Define Subject as
    [subject: [cat: np]]

Define Object as
    [object: [cat: np]]

Define Intransitive as
    Verb
    Subject

Define Transitive as
    Intransitive
    Object

Define SubjectPossObjectAgreement as
    [subject: [agr: $1]
    object: [poss: [agr: $1]]]

Define ZijnKansenWaarnemen as
    Transitive
    SubjectPossObjectAgreement
    [lex: waarnemen
     object: [lex: kans
              agr: [num: pl]
              modifiable: -]
    sem: [paraphrase: seize_the_opportunity]]

Word waarnemen:
    Verb
    {[WaarnemenLiteral] [ZijnKansenWaarnemen]}
    [lex: waarnemen]
```

Figure 4: PATR-II description (with PC-PATR notation) of (NL) *zijn kansen waarnemen* 'to seize the opportunity'

```
Define ImpersIntransitive as
    [cat: v
     pers: 3
     num: sg
     subject: -
     object: -]

Define IndirectObject as
    [iobject: [cat: np
               case: dat]]

Define PrepositionalObject as
    [pobject: [cat: pp]]

Define DobrzeZOczuPatrzy as
    ImpersIntransitive
    IndirectObject
    PrepositionalObject
    Adverb
    [pobject: [lex: z
               object: [cat:np
                        case: gen
                        num: pl
                        lex: oko
                        modifiable: -]]
     adverb: [word: dobrze
              position: initial]]
     sem: [paraphrase: someone_looks_like_a_good_person]

Word patrzy:
    Verb
    {[PatrzecLiteral] [DobrzeZOczuPatrzy]}
    [lex: patrzeć]
```

Figure 5: PATR-II description (with PC-PATR notation) of (PL) *dobrze [KOMUŚ] z oczu patrzy* ‘someone looks like a good person’

time. Remember that the constraints on the occurrence of certain arguments can be encoded by using subcategorization lists in the usual way. This is left out in the example. Now, compared to the Walenty encoding in Figure 3, the corresponding PART-II template `DobrzeZOczuPatrzy` is much more verbose, not only because it contains more information. But this should not be taken as a general disadvantage, as it can help to promote transparency.

Summing up, the examples provided here demonstrate that PATR-II does many important things right: it makes available a transparent, flexible enough encoding language; it has a well-defined denotational semantics; it includes means to arbitrarily factorize properties and to express generalizations even beyond strict monotonicity. In our view, this makes PATR-II better suited to encode MWEs than DuELME and Walenty *in the long run*, since it can integrate unforeseeable properties, regularities or encoding styles much easier.

Yet at the same time, encoding with PATR-II is subject to some severe restrictions:

- PATR-II does not seem to allow for templates to be embedded. Hence, templates can only be applied to the root of a feature structure description.
- Feature structures are untyped in PATR-II which makes them harder to be checked for consistency or to encode representations that rely on types.
- PATR-II allows one to describe full word forms as terminals of CFG rules, but it is not possible to analyze them further, that is, describe the underlying morphemes and how they combine. Consequently, it is at least tedious to describe morphological paradigms. This is something that, for example, DATR (Evans & Gazdar 1996) is better suited for.
- In PATR-II, word order constraints are accounted for by filtering CFG rules via features. Thus, it is not possible to state these constraints in just one place, but one has to think of which features prohibit or trigger the application of which CFG rules in which situation of a derivation.

Furthermore, as we said before, PATR-II chooses a lexical approach to argument structure in the sense of Müller & Wechsler (2014) where the argument structure emerges from lexical units and crucially determines the syntax. The other extreme, namely the phrasal approach to argument structure, rather puts emphasis on the syntactic side, assuming phrasal representations of argument structure that exist independently of lexical anchors. This latter approach better fits into the encoding format of XMG, which will be presented next.

7.2 eXtensible MetaGrammar

The framework of eXtensible MetaGrammar (XMG, Crabbé et al. 2013 and XMG2, Petitjean et al. 2016) most obviously differs from the ones of PATR-II, DuELME and Walenty in that it can be used to generate a wide range of linguistic resources. The variety of these resources is made possible by XMG's modularity and extensibility, allowing to create new dedicated compilers using adapted description languages. XMG is a multi paradigm language, as it manipulates programs (metagrammars) which make intensive use of logic (such as Prolog programs) and constraints. XMG also borrows some aspects from object-oriented programming, whose advantages in the context of linguistic knowledge description are discussed in Daelemans & De Smedt (1994). The most obvious example of such an aspect is that XMG descriptions are organized into CLASSES, which have encapsulated name spaces. Inheritance relations may hold between classes, and the scope of the identifiers is explicitly controlled, thanks to `export` statements. The crucial elements of a class are DIMENSIONS. Each of them is equipped with a description language, which is specifically adapted to the kind of structures needed in the dimension (trees, predicates, …). Dimensions are compiled independently, thereby enabling the grammar writer to treat the levels of linguistic information separately. In the following, we will be using the dimension `<syn>` for the syntax and the more recent `<frame>` dimension for frame-semantic descriptions, skipping over other available dimensions. Note that `<syn>` contains tree descriptions where nodes may carry untyped feature structures, while `<frame>` comprises *typed* feature structure descriptions (Lichte & Petitjean 2015).

Figure 6 shows a part of a tentative XMG encoding of (NL) *zijn kansen waarnemen*. The first thing to notice when comparing the XMG description to the DuELME counterpart in Figure 2 is that there is no principled distinction between "patterns" and "MWE descriptions" (similarly to the PATR-II encoding in Figure 4). Rather, they are equally represented as classes, yet of varying specificity. Crucially, the classes stand in inheritance relations, here marked with the `import` statement. For example, the most basic class shown in Figure 6, `intransitive[]`, imports two other classes, `subject[]` and `verb[]` (cf. Line 2). On the other hand, `intransitive[]` is further handed down to `transitive[]`, just adding `object[]`. Finally, `transitive[]` is imported into `subject_poss_object_agreement[]` to add the compulsory agreement between the subject and the possessive pronoun of the object, and, in turn, this class is further imported into `zijn_kansen_waarnemen[]`, which is the class of the MWE proper. Hence, `subject_poss_object_agreement[]` contains the more regular properties of the MWE, and `zijn_kansen_waarnemen[]` the less regular ones. The corresponding inheritance hier-

```
class intransitive
import subject[] verb[]
{ <syn> { ?Subj >>+ ?V }}

class transitive
import intransitive[] object[]
{ <syn> { ?Subj >>+ ?Obj;
          ?Obj >>+ ?V } }

class subject_poss_object_agreement
declare ?Subj ?Obj ?NUM ?PERS ?GEND
export ?Subj ?Obj
{ <syn> {
    ?Subj[num=?NUM,pers=?PERS,gend=?GEND];
    ?Obj [] {
      [cat=d,num=pl,possnum=?NUM,pers=?PERS,gend=?GEND] "zijn"}}}

class zijn_kansen_waarnemen
import transitive[] subject_poss_object_agreement[]
declare ?I
{ <syn> {
    ?Subj[i=?I];
    ?Obj [] {
      [cat=n,modifiable=-,num=pl] "kans"};
    ?V[] "waar_nehmen" };
  <frame> {
     [using-event,
      actor:?I,
      theme:chance]}}
```

Figure 6: XMG encoding of *zijn kansen waarnemen* ('to seize the opportunity')

archy of the used classes is shown in Figure 7, in which the MWE shows up as leaf, i.e. as the most specific class. Note that this inheritance hierarchy mirrors the one of the PATR-II encoding in Figure 4.

In general, classes that correspond to irregular or weakly regular properties of lexical entries appear as leaves, whereas more regular aspects are assigned to dominating classes. Hence, "patterns" can be arbitrarily factorized, which is in sharp contrast to the DuELME encoding format. Another difference is the general availability of variables in XMG, which are commonly prefixed with a question mark. This is exploited in `subject_poss_object_agreement[]` when expressing

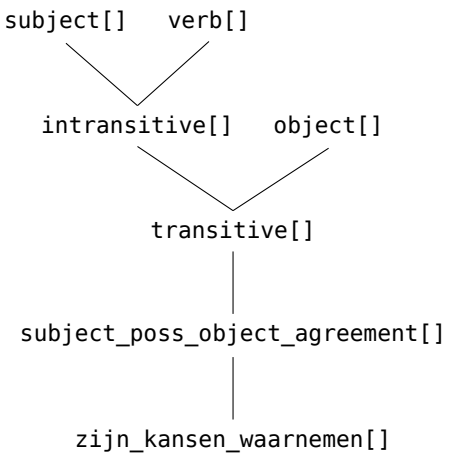

Figure 7: Inheritance hierarchy of XMG classes according to the code in Figure 6

agreement between the subject and the possessive determiner using the variables `?NUM`, `?PERS`, and `?GEND` (cf. Lines 14 and 16). Variables are also used for sharing information between dimensions, for example between `<syn>` and `<frame>`, which holds the idiomatic meaning of the MWE, in class `zijn_kansen_waarnemen[]`: the unification variable `?I` here is the frame referent of the subject, and consequently appears both in the syntactic node `?Subj` and as the value of the feature `actor` in the semantic frame. Finally, features and variables can be freely added to XMG, for example, features to indicate constraints on modification (`modifiable`) or passivization.

Remember that the descriptions in `<syn>` are tree descriptions, which are able to express the usual, potentially underspecified node relations regarding dominance and precedence. For example, `>>+` (cf. Lines 3, 7 and 8 in Figure 6) expresses the transitive, non-reflexive precedence relation between two nodes of a tree. As the tree descriptions can be underspecified in this way, the denotation can be a set of trees. XMG comes with a solver for these descriptions, and a viewer, both of which are available online.[14] Hence, the solutions can be inspected independently of a specific application belonging to some specific framework.

The preliminary XMG encoding of (PL) *dobrze [KOMUS] z oczu patrzy* is presented in Figure 8.[15]

[14] http://xmg.phil.hhu.de/

[15] We owe the frame semantic representation in Figure 8 to Rainer Osswald.

```
class impers_intransitive
export ?VP ?V
declare ?VP ?V
{ <syn>{
    ?VP [cat=vp] { ?V [cat=v,pers=3,num=sg] }}}

class dobrze_z_oczu_patrzy
declare ?I ?A ?P
import impers_intransitive[] ind_object[] pp_object[] adverb[]
{ <syn> {
    ?IndObj [i=?I];
    ?AdvP [] { ?A [] "dobrze"};
    ?PP [] { [case=gen] "z"
      [] {
        [num=pl,modifiable=-] "oko"}};
    ?V "patrzeć";
    ?VP -> ?PP;
    ?VP -> ?IndObj;
    ?AdvP >>+ ?PP;
    ?AdvP >>+ ?V };
  <frame> {
    [impression-about,
     perceiver: ?P,
     theme: ?I,
     content:[has-prop,
              theme: ?I,
              prop: good]
    ]}
}
```

Figure 8: XMG encoding of *dobrze* [KOMUŚ] *z oczu patrzy* ('someone looks like a good person')

Again, the class that corresponds to the MWE, `dobrze_z_oczu_patrzy[]`, inherits from more abstract (and "regular") classes, which can be also seen from the inheritance hierarchy in Figure 9.

Here, the `impers_intransitive[]` class encodes the fact that the subject is absent (as only the verb phrase and its subordinate verb are listed), and that the (impersonal) verb must occur in the third person singular. Finally, the `dobrze_z_oczu_patrzy[]` class reuses the previous class and adds the compulsory adverb. Moreover, certain nodes, identified by shared variables, are further specified for lemmas (in double quotes) and all weakly regular morphological constraints are

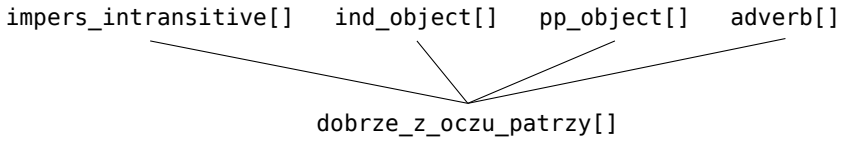

Figure 9: Inheritance hierarchy of XMG classes according to the code in Figure 8

listed. Notably, the noun governed by the preposition *z* 'from' is restricted to the lemma `oko` 'eye' and to plural, and its modification is prohibited. Note that the genitive case of *oko* is not specified in this class, as it is already part of the agreement rules which were inherited from the `pp_object[]` class. Linearization constraints on the adverb appear in Lines 19–20. The example also includes dominance constraints in Lines 17–18 that use `->` to describe an immediate dominance relation. Finally, we use unification variable once again to express the fact that the semantic referent of the syntactic subject (`?I`) is the theme of the semantic frame of the MWE. This frame can be read as follows: a perceiver `?P`, left unspecified, has an impression about `?I`, and this impression is that `?I` has the property of being a good person. Thus, all the necessary constraints imposed on this MWE can be covered at various abstraction levels, while factorizing information in such a way that the `dobrze_z_oczu_patrzy[]` class only contains the constraints which are specific to the MWE or at least weakly regular.

By way of conclusion, let us compare the presented encoding examples for PATR-II and XMG in more detail. Despite their large commonalities when contrasting them with fixed encoding formats such as DuELME and Walenty, PATR-II and XMG can differ considerably in some of their properties.

- In the given examples, XMG is constructionist in the sense that it models phrasal units, whereas PATR-II assumes a head-driven (or "lexicalist", Müller & Wechsler 2014) approach to representing argument structure. However, this is not to say that XMG cannot be also used in a head-driven way.

- XMG supports type inferences, hence the unification of typed feature structures. In PATR-II, feature structures are strictly untyped.

- XMG comes with different description languages as well as different types of models, namely trees, typed feature structures, expressions of predicate logic and even strings. PATR-II is restricted to the description of feature structures and CFG rules.

- XMG allows for directed inheritance in the sense that inherited descriptions can be added to any part of the description, not just the root part as with PATR-II.
- XMG is more verbose than PATR-II because it is designed to implement a truly object-oriented programming style with encapsulated namespaces etc. When considering just toy examples, it is admittedly just a matter of taste whether this is something worthwhile. In large-scale grammars and lexicons, however, the advantage can be more substantial by helping to ensure consistency due to the extra checking done by the solver.

In sum, XMG seems to be generally more powerful than PATR-II, but also more cumbersome in the way of encoding.

8 Summary

Table 2 shows a comparison of the encoding formalisms presented in Sections 6 and 7 with respect to the encoding virtues described in Sections 4.2 and 4.3. We omit the encoding virtues with respect to a lexical object (cf. Section 4.1). They are mostly related to a particular lexical encoding and not to the underlying formalism.

Table 2: Ranking of encoding formats in different categories – lexical encoding virtues – with special focus on MWEs. The range of values is from 1 to 4, where 1 means that we judge the corresponding format as relatively the best in the given category.

	human user oriented				lexical resource oriented
	TRANSPARENCY	FLEXIBILITY	POWER TO GENERALIZE	IMPLEMENTATION FRIENDLINESS	ELECTRONIC VERSATILITY
DuELME	4	4	3	2	1
Walenty	3	3	3	1	1
PATR-II	1	2	2	4	4
XMG	1	1	1	3	3

Descriptions in PATR-II and XMG come with clear denotational semantics, which makes these two formalisms stand out as highly transparent in comparison with their less flexible counterparts. Transparency of the Walenty's encoding format is relatively high. Due to its conciseness, it is possible to read, analyze

and write new entries relatively quickly. However, this requires some experience, since interpretation of certain syntactic constructions (e.g., positions in lexically restricted phrase descriptions) is implicit. More importantly, interpretation of the meaning of symbols used in Walenty descriptions is often implicit as well. Certain patterns – for instance, a prepositional noun phrase (PREPNP) – are defined as atomic constructions, and the recommended way to model new phenomena – for instance, agreement between the subject and the possessive determiner of the direct object – is to add new symbols to the alphabet of the formalism.[16] This can be seen as a flexible solution, but it may also lead to proliferation of atomic symbols with encoding-specific semantics, not defined within the formalism itself. This in turn may harm transparency of the individual Walenty-based encodings and decrease its overall electronic versatility. Finally, there seems to be no clear denotational semantics defined for DuELME descriptions (except, maybe, in its LMF standard export format). Their interpretation is based partially on formal properties and inference rules, partially on methodological recommendations, and the borderline between the two is hard to determine, which severely harms the clarity of the format.

Not very surprisingly, XMG and PATR-II are also more flexible than Walenty or DuELME. In comparison to XMG, PATR-II exhibits certain restrictions (see Section 7.1 for details) which limit, among others, its power to express word order constraints.[17] Walenty is flexible enough to account for most of the MWE-related properties. Yet, the need to introduce new symbols to express previously unforeseen phenomena (already mentioned w.r.t. the virtue of transparency) may stem from the insufficient flexibility of the formalism. As for DuELME, we see its relatively low transparency as the main cause of its relatively low flexibility – it is hard to define complex constructions when clear foundations are not established.

The restrictions enforced by PATR-II diminish also its power to express certain factorizations – notably, by not allowing templates to apply to feature structure nodes other than roots. Due to the untyped nature of feature structures, representation of certain properties based on types – and, therefore, the related generalizations – may be hindered as well. The power to generalize of DuELME is limited by the distinction between patterns and MWE descriptions. Moreover, DuELME provides no way to express any kind of sharing between the individual patterns. As to Walenty, a hierarchy of macros (in the sense that a macro can

[16] In fully flexible formalisms such new syntactic phenomena can be factored through the use of dedicated classes whose semantics remains explicit.

[17] Note, however, that while word order constraints are supposed to be expressed in PATR-II through filtering CFG rules via features, these constraints could be also expressed directly as feature structure values.

refer to other macros) can be used to account for repeating patterns. However, it is not clear to what extent macros constitute a part of the formalism itself and it seems that the mechanism of macros is too simple to account for more complex patterns (for example, the abovementioned subject/possessive agreement restriction).

Both DuELME and Walenty seem to be more electronically versatile than XMG. DuELME supports the standard LMF format, while one of the formats supported by Walenty is TEI – based on XML, less concise than the default Walenty's format but more explicit and application-friendly. While XMG encodings can be compiled and stored in an XML format which directly represents all the resolved property names, it does not necessarily contain all the underlying generalizations (i.e., those encoded in the class inheritance hierarchy). One could imagine parsing and interpreting XMG descriptions themselves, and not the resulting compiled encodings, as a first step of converting XMG descriptions to a particular lexical resource. However, this solution would require certain knowledge about the formal principles and mechanisms underlying XMG. Thus the additional flexibility and power to generalize of XMG come with additional cost in terms of the preprocessing work that needs to be done to obtain a particular resource from XMG descriptions. As to PATR-II, there seem to be very few actively maintained software tools for it. While a parser of this formalism can still be downloaded, its further development has been discontinued as of 2006.[18] We therefore estimate the electronic versatility of PATR-II as being rather low due to the current unavailability of dedicated software tools.

Implementation friendliness of DuELME and Walenty has been already confirmed in practice. DuELME has been used to encode a lexicon of 5,000 Dutch MWEs, while Walenty underlies The Polish Valence Dictionary which, in particular, contains around 8,000 MWE entries. Moreover, a dedicated tool Slowal (http://zil.ipipan.waw.pl/Slowal) has been designed for creating, editing and browsing Walenty dictionaries. Thus, Walenty comes with an implementation friendly environment, editing tools and, on top of that, provides conversion between several dictionary formats adapted for different needs. In XMG, MWEs are defined as terminal classes and are encoded directly in the source code. At the moment, there is no dedicated tool which would assist a human user with encoding large sets of MWEs. At the same time, encoding MWEs directly in the source code can be seen as a flexible solution which allows the user to adopt his or her own organization of MWE-related classes. High factorization capabilities of XMG should also facilitate handling large sets of lexical objects, heterogeneous yet often showing

[18] http://software.sil.org/pc-patr/

common patterns. On top of that, the process of compiling XMG descriptions provides a verification mechanism which allows to check the correctness of the individual XMG-based lexical entries. For PART-II, again, we found no readily available software tool that is designed to support the implementation process.

As a general conclusion, lexical encoding of MWEs is a highly challenging task, as also stressed in Dyvik et al.; Markantonatou et al. (2019; 2019 [this volume]), due to the complexity and versatility of the regular and idiosyncratic phenomena exhibited by the linguistic objects. The four encoding formats examined here show complementary strengths and weaknesses. We believe that transparency, flexibility and the power to generalize[19] are the fundamental virtues to promote in lexical encoding of MWEs, and in this respect XMG seems to stand out as a particularly appropriate framework. These qualities have to be confirmed, however, in large-scale lexicographic efforts, which call for enhancing its implementation friendliness via developing a lexicographic framework to automate the encoding and validation process. Note finally that relatively few considerations have been made here on semantic properties of MWEs. Maybe the most outstanding feature of many MWEs is their semantic non-compositionality, and addressing it in a lexical encoding framework remains one of the most challenging perspectives.

Acknowledgements

This work has been supported by the IC1207 PARSEME COST action, by the Deutsche Forschungsgemeinschaft (DFG) within the CRC 991 "The Structure of Representations in Language, Cognition, and Science", and by a doctoral grant from the French Ministry of Higher Education and Research.

References

Copestake, Ann. 2002. *Implementing typed feature structure grammars*. Stanford: CSLI Publications.

Crabbé, Benoît, Denys Duchier, Claire Gardent, Joseph Le Roux & Yannick Parmentier. 2013. XMG: eXtensible MetaGrammar. *Computational Linguistics* 39(3). 591–629. DOI:10.1162/COLI_a_00144

Daelemans, Walter & Koenraad De Smedt. 1994. Default inheritance in an object-oriented representation of linguistic categories. *International Journal of Human-Computer Studies* 41(1). 149–177.

[19]We believe that the latter property – the power to generalize – should be particularly helpful in modeling the varying degrees of flexibility exhibited by MWEs, discussed in Sheinfux et al. (2019 [this volume]).

Daelemans, Walter & Erik-Jan van der Linden. 1992. *Evaluation of lexical representation formalisms*. ITK Research Memo. Tilburg: Institute for Language Technology & Artificial Intelligence, Tilburg University.

Dyvik, Helge, Gyri Smørdal Losnegaard & Victoria Rosén. 2019. Multiword expressions in an LFG grammar for Norwegian. In Yannick Parmentier & Jakub Waszczuk (eds.), *Representation and parsing of multiword expressions: Current trends*, 69–108. Berlin: Language Science Press. DOI:10.5281/zenodo.2579037

Evans, Roger & Gerald Gazdar. 1996. DATR: A language for lexical knowledge representation. *Computational Linguistics* 22(2). 167–216.

Francopoulo, Gil, Monte George, Nicoletta Calzolari, Monica Monachini, Nuria Bel, Mandy Pet & Claudia Soria. 2006. Lexical markup framework (LMF). In *Proceedings of the international conference on Language Resources and Evaluation (LREC 2006)*, 233–236.

Grégoire, Nicole. 2007a. *MWE Lexicon for Dutch: Encoding protocol*. http://duelme.clarin.inl.nl/documentation.php.

Grégoire, Nicole. 2007b. *MWE Lexicon for Dutch: Overview of pattern descriptions*. http://duelme.clarin.inl.nl/documentation.php.

Grégoire, Nicole. 2010. DuELME: A Dutch electronic lexicon of multiword expressions. *Language Resources and Evaluation* 44(1–2). 23–39.

Gross, Maurice. 1986. Lexicon-grammar: The representation of compound words. In *Proceedings of the 11th coference on Computational Linguistics* (COLING '86), 1–6. Bonn, Germany: Association for Computational Linguistics.

Gross, Maurice. 1994. Constructing lexicon-grammars. In B.T.S. Atkins & A. Zampolli (eds.), *Computational Approaches to the Lexicon*, 213–263. Oxford: Oxford University Press.

Habert, Benoît & Christian Jaquemin. 1995. Construction nominales à contraintes fortes et grammaires d'unification. *Lingvisticae Investigationes* 19(2). 401–427.

Kilbury, James, Petra Naerger & Ingrid Renz. 1991. DATR as a lexical component for PATR. In *Proceedings of the fifth conference of the European Chapter of the Association for Computational Linguistics (EACL-91)*, 137–142.

Lichte, Timm & Simon Petitjean. 2015. Implementing semantic frames as typed feature structures with XMG. *Journal of Language Modelling* 3(1). 185–228.

Markantonatou, Stella, Niki Samaridi & Panagiotis Minos. 2019. Issues in parsing MWEs in an LFG/XLE framework. In Yannick Parmentier & Jakub Waszczuk (eds.), *Representation and parsing of multiword expressions: Current trends*, 109–126. Berlin: Language Science Press. DOI:10.5281/zenodo.2579039

McConnel, Stephen. 1997. *PC-PATR Reference Manual*. Summer Institute of Linguistics. Dallas, TX. http://www.sil.org/pcpatr/manual/pcpatr.html. Version 0.99b5.

Mel'čuk, Igor, Nadia Arbatchewsky-Jumarie, Louise Dagenais, Léo Elnitsky, Lidija Iordanskaja, Marie-Noëlle Lefebvre & Suzanne Mantha. 1988. *Dictionnaire explicatif et combinatoire du français contemporain: Recherches lexico-sémantiques*. Vol. II (Recherches lexico-sémantiques). Presses de l'Univ. de Montréal.

Müller, Stefan & Stephen M. Wechsler. 2014. Lexical approaches to argument structure. *Theoretical Linguistics* 40(1–2). 1–76.

Nitoń, Bartłomiej, Tomasz Bartosiak & Elżbieta Hajnicz. 2016. Accessing and elaborating Walenty – a valence dictionary of Polish – via Internet browser. In *Proceedings of the 10th edition of the Language Resources and Evaluation conference (LREC)*, 1352–1359. Portorož, Slovenia.

Odijk, Jan. 2013. DUELME: Dutch electronic lexicon of multiword expressions. In Gil Francopoulo (ed.), *LMF: lexical markup framework*, chap. 9, 133–144. Wiley-ISTE.

Petitjean, Simon, Denys Duchier & Yannick Parmentier. 2016. XMG2: Describing description languages. In Maxime Amblard, Philippe de Groote, Sylvain Pogodalla & Christian Retoré (eds.), *Logical aspects of computational linguistics: Celebrating 20 years of LACL (1996–2016)*, 255–272. Berlin & Heidelberg: Springer.

Przepiórkowski, Adam, Jan Hajič, Elżbieta Hajnicz & Zdeňka Urešová. 2016. Phraseology in two Slavic valency dictionaries: Limitations and perspectives. *International Journal of Lexicography* 29.

Przepiórkowski, Adam, Elżbieta Hajnicz, Agnieszka Patejuk & Marcin Woliński. 2014. Extended phraseological information in a valence dictionary for NLP applications. In *Proceedings of the workshop on Lexical and Grammatical Resources for Language Processing (LG-LP 2014)*, 83–91. Dublin, Ireland.

Savary, Agata. 2008. Computational inflection of multi-word units: A contrastive study of lexical approaches. *Linguistic Issues in Language Technology* 1(2). 1–53.

Sheinfux, Livnat Herzig, Tali Arad Greshler, Nurit Melnik & Shuly Wintner. 2019. Verbal multiword expressions: Idiomaticity and flexibility. In Yannick Parmentier & Jakub Waszczuk (eds.), *Representation and parsing of multiword expressions: Current trends*, 35–68. Berlin: Language Science Press. DOI:10.5281/zenodo.2579035

Shieber, Stuart M. 1984. The design of a computer language for linguistic information. In *Proceedings of the 10th international conference on Computational Linguistics and 22nd annual meeting of the Association for Computational Linguistics (ACL 1984)*, 362–366. Stanford, CA. http://www.aclweb.org/anthology/P84-1075.

Shieber, Stuart M. 1986. *An introduction to unification-based approaches to grammar* (CSLI Lecture Notes Series 4). Stanford, CA: CSLI.