

Uniwersytet Warszawski
Wydział Neofilologii

Radosław Moszczyński

Nr albumu: 196457

Formal approaches
to multiword lexemes

Praca magisterska
na kierunku FILOLOGIA
w zakresie FILOLOGIA ANGIELSKA

Praca wykonana pod kierunkiem
dra hab. Janusza S. Bienia, prof. UW
Katedra Lingwistyki Formalnej

Maj 2006

Oświadczenie kierującego pracą

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełnia ona warunki do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpis autora (autorów) pracy

Streszczenie

Formalne opisy leksemów wielowyrazowych

Praca omawia zagadnienia leksemów wielowyrazowych m. in. z punktu widzenia kompilacji słowników elektronicznych. Pierwsza część pracy skupia się na teoretycznym opisie leksemów wielowyrazowych z naciskiem na ich zmienność leksykalną i składniową. Część druga dotyczy sposobów przetwarzania leksemów wielowyrazowych na potrzeby słowników elektronicznych. W szczególności omówiony jest formalizm IDAREX oraz jego dotychczasowe zastosowania. Ostatni rozdział zawiera wstępną specyfikację obsługi zbliżonego formalizmu w protokole DICT.

Słowa kluczowe

leksem wielowyrazowy, kolokacja, słownik elektroniczny, IDAREX, DICT

Dziedzina pracy (kody wg programu Socrates-Erasmus)

9.3 Lingwistyka

Contents

1. Introduction	4
1.1. Aims	4
1.2. Synopsis	4
2. Overview of multiword lexemes	6
2.1. Terminology and definitions	6
2.2. Classification	9
2.2.1. Formal grammar approach	10
2.2.2. NLP approach	11
2.3. Lexical and syntactic variability of multiword lexemes	12
2.3.1. Lexical operations	13
2.3.2. Syntactic transformations	14
2.3.3. Internal and external transformations	15
2.3.4. Explaining multiword lexeme variability	15
2.3.5. A practical approach to describing variability	19
3. Multiword lexemes in printed dictionaries	20
3.1. Selection	21
3.2. Identification	22
3.3. Lexicographic practice	24
3.3.1. General-purpose dictionaries	24
3.3.2. Specialized dictionaries	27
4. Computational approach to multiword lexemes	29
4.1. Regular expressions	29
4.1.1. Basic concepts	29
4.1.2. Regular expressions and natural languages	30
4.1.3. Finite state machines	31
4.2. IDioms As REgular eXpressions (IDAREX)	31
4.2.1. Xerox's regular expressions	32
4.2.2. Two-level morphology	33
4.2.3. IDAREX operators and macros	34
4.3. Practical applications of IDAREX	37
4.3.1. LOCOLEX	37
4.3.2. COMPASS	39
4.3.3. STEEL	40
4.4. Similar applications	43
4.4.1. MoBiMouse	43

5. Towards a free implementation of context-sensitive dictionaries . . .	45
5.1. Extending the DICT protocol	45
5.1.1. DICT — a dictionary server protocol	45
5.1.2. Multiword lexemes and DICT	47
5.2. Reusing Piotrowski and Saloni’s dictionary	49
5.3. Verifying a multiword lexeme formalism on a corpus	49
6. Conclusions	51

List of Figures

2.1. Compositionality of meaning in idiomatic phrases and idiomatically combining expressions.	17
3.1. Zgusta's classification of multiword units.	22
4.1. A sample entry from Saloni and Piotrowski's dictionary encoded in \TeX	41
4.2. An entry in the printed version of Saloni and Piotrowski's dictionary.	41
4.3. A sample, incomplete SGML entry from Saloni and Piotrowski's dictionary containing an IDAREX expression.	42
4.4. MoBiMouse displays its stemming capabilities.	44

Chapter 1

Introduction

1.1. Aims

Multiword lexemes have always been problematic for both theoretical and applied linguistics. Difficulties at the most basic level stem from the fact that such units are hard to define in an unambiguous manner. Moreover, researchers often do not agree on the way in which they should be classified, which results in a confusing multitude of taxonomies whose individual subsets often overlap each other.

To make matters worse, problems multiply on higher levels. Formal linguists find it hard to construct logically sound models for describing multiword lexemes because they are variable and include a wide range of phenomena, from simple compounds to sentence patterns. For lexicographers, the main problem lies in establishing which multiword lexemes are “frozen” enough to be included as separate headwords in dictionaries. Researchers from the field of natural language processing find it hard to come up with language processing tools that are able to adequately cover multiword lexemes, mainly because of difficulties associated with recognizing them. All this is just to signal the most important problems, this list could go on for much longer.

The aims of this thesis are threefold. Firstly, it tries to present the current state of knowledge about multiword lexemes from the point of view of formal linguistics. Secondly, it shows how multiword lexemes can be formalized for the purposes of lexicography and electronic dictionaries. Thirdly, it suggests some ideas for implementing and evaluating a multiword lexeme formalism, most importantly by means of an extension to the DICT protocol.

1.2. Synopsis

The structure of the thesis is as follows (excluding the current chapter):

Chapter 2 introduces the notion of multiword lexemes. It presents their definitions and classifications, and summarizes the current state of knowledge about their syntactic and semantic properties focusing on their variability.

Chapter 3 discusses the issue of including multiword lexemes in traditional printed dictionaries.

Chapter 4 presents the IDAREX (IDioms As REgular eXpressions) formalism and some of its practical applications.

Chapter 5 suggests integrating a similar formalism with the DICT protocol and verifying it on a corpus.

Chapter 6 contains conclusions and ending remarks.

Chapter 2

Overview of multiword lexemes

Second language learners, and also computer systems that generate natural language, often produce expressions that native speakers find either ugly or simply wrong. In many cases this results from inappropriate usage or failure to use multiword lexemes, i.e. complex linguistic units that function as semantic wholes, whose behavior and properties cannot be predicted on the basis of grammar rules and lexicon entries alone. This is because most multiword lexemes are not fixed and can undergo transformations of varying complexity. Moreover, they often violate grammatical rules or contain words that do not function anywhere else in the language. A careful analysis of multiword lexemes inevitably leads to difficult higher-level questions concerning the boundary between literal and figurative language, as well as the extent of meaning compositionality. As this thesis will show, the properties of multiword lexemes make them difficult to handle for formal linguists, natural language processing engineers, and lexicographers alike.

The aim of this chapter is to present some theory upon which multiword lexeme studies are based. Section 2.1 contains terminological preliminaries and tries to establish the most accurate definition of multiword lexemes based on the ones scattered throughout linguistic literature. Section 2.2 contains examples of multiword lexeme taxonomies. Finally, Section 2.3 discusses the most interesting and problematic property of multiword lexemes, which is their variability.

2.1. Terminology and definitions

The notion of multiword lexemes¹ is not clear. The first difficulty one has to face is the broad range of linguistic units that it covers, which seemingly have little in common:

*San Francisco, ad hoc, fresh air, telephone box, part of speech, take a walk,
call (somebody) up, pull strings, keep tabs on somebody, spill the beans,
kick the bucket*

The second one is terminology. Following is just a sample of terms that describe linguistic phenomena that can be referred to as multiword lexemes:

¹Terminology concerning this linguistic phenomenon is not stable. At this point it should be assumed that the term is used in the broadest, generic sense whose approximate meaning is “units of a language’s lexical system composed of several separated words”.

idiom, collocation, compound word, fixed syntagm, phraseologism, lexical solidarity, phraseolexeme, polylexical expression, multiword expression, multiword lexeme, multiword unit, multilexemic expression, multiple-word expression

Sometimes there is no practical difference between the terms, e.g. one can safely assume that a *multiword lexeme* and a *multiword expression* are the same things, at least in the writings of linguists belonging to the same circles. However, in most cases there are differences as far as scope of the terms is concerned. To some researchers the units in question are institutionalized phrases whose meaning cannot be inferred on the basis of the meanings of the words that constitute them. To others they are any units that consist of more than one word and follow some formal regularity (regardless of whether their meaning is literal or metaphorical). The boundary seems to be one between areas of linguistic research² — lexicographers are more interested in fixed phrases whose meaning is not obvious and which should be treated as separate entries in dictionaries, whereas NLP³ researchers are interested in all patterns that computers can recognize and generate, even the ones which human language users find to be obvious and mostly uninteresting.

The term *multiword lexeme* has been chosen for the purposes of this thesis because it was used by some researchers working on the IDAREX formalism, to which a large part the thesis is devoted. Before establishing a working definition of the term, I will continue to use it in the generic sense mentioned earlier.

There exist several definitions of multiword lexemes. Some very old ones can be found in Smith 1943. In this view multiword lexemes are expressions “peculiar to a people or nation” or “phrases which are verbal anomalies, which transgress ... either the laws of grammar or the laws of logic.”

Another definition, which was preferred by the generative school, says that a multiword lexeme is a “collection of words whose meaning as a whole cannot be derived from the meanings of the individual words” (Reagan 1987, 417). Another way to put it is to say that a multiword lexeme is “something a language user could fail to know while knowing everything else in the language” (Fillmore, Kay, and O’Connor 1988, 504). This assumes meaning non-compositionality which, as it will be shown in Section 2.3, is not entirely right. Despite its inaccuracy, this is the most commonly quoted definition, probably because of its long tradition and the fact that for most purposes that do not require an in-depth analysis it is adequate.

It has been pointed out that this definition is circular (Cruse 1986, 37). It seems that it is not clear whether the meaning of the whole unit cannot be inferred from the meanings its parts carry *in this very unit* or in other expressions. The former is excluded by the definition. The latter, however, is equally problematic: while considering *to pull someone’s leg* it is useless to refer to *leg* in *He hasn’t a leg to stand on*. If the definition were to be sound, it should specify that a multiword lexeme is an expression whose meaning cannot be inferred from the meanings its parts have *when they are not parts of other multiword lexemes*. This is where circularity shows: before one can use this definition to determine whether an expression is a multiword lexeme, he already has to be able to distinguish idiomatic from non-idiomatic expressions.

Other researchers follow a slightly different path and do not refer to compositionality of meaning. In this view a multiword lexeme should “consist of more than one lexical

²This was suggested by prof. J.S. Bień.

³NLP — natural language processing.

constituent” and function as a “single minimal semantic constituent” (Cruse 1986, 37). A similar definition is provided by Reagan (1987, 418). In the author’s view, multiword lexemes are “all linguistic units composed of two or more words, which semantically function as wholes.”

Yet another approach can be found in Nunberg, Sag, and Wasow 1994. Instead of providing a direct definition, the authors propose a set of properties on the basis of which one should distinguish multiword lexemes (or *idioms*, as they call them) from regular linguistic units. In their view, multiword lexemes are characterized by the following:

1. Conventioanality — the meaning and usage of multiword lexemes cannot be entirely predicted on the basis of conventions that determine the usage of their constituents when they are used in isolation.
2. Inflexibility⁴ — unlike common expressions, multiword lexemes typically appear only in a limited number of syntactic constructions. Many transformations, such as passivization, make them lose their idiomatic meaning (*the bucket was kicked by John* does not refer to dying).
3. Figuration — there is a subset of multiword lexemes that often involves metaphors (*take the bull by the horns*), metonymies (*lend a hand*), and other devices typical of figurative language. It should be noted that speakers may often not know the reasons why some particular metaphor is used in a given case, e.g. why *shoot the breeze* should mean ‘to chat’, but in general they do perceive that some figuration is involved.
4. Proverbiality — multiword lexemes are often used to describe and explain recurring social situations in terms of concrete things and relations (*chew the fat, spill the beans*).
5. Informality — some types of multiword lexemes are most frequently used in informal registers and speech.
6. Affect — multiword lexemes are often used to express an evaluation or personal emotions concerning the things they refer to.

Among all these properties, only conventionality is considered to be a required one, and all the others are optional. Thanks to this, the set of properties can adequately describe both idioms in the traditional sense of Smith 1943 and syntactic patterns without any metaphorical meaning, sought by computational linguists.

It seems that the most useful definition, in the sense that it covers the broadest range of phenomena, can be found in Brundage et al. 1992. This is precisely the definition which has been used earlier and referred to as “generic”, according to which multiword lexemes are “units of a language’s lexical system (= lexemes) composed of several separated words (=multiword)”. This will be the working definition for the rest of the current thesis. In its authors’ view, it is lexical properties that distinguish multiword lexemes from regular constructions:

⁴In the light of the whole article, this property should be probably called “limited flexibility”.

The first important property of an MWL is its lexeme status. This is the essential property in which MWLs differ from free syntagmatic constructions that are produced according to syntactic structure models every single time they are stated. On the contrary, MWLs are lexicalized and therefore reproduced as lexical units of the language system in question. (Brundage et al., 4)

Apart from that, the authors list the following properties which all multiword lexemes are characterized by⁵:

1. Semantic compositionality — typical free constructions display this property, whereas typical multiword lexemes do not (this is problematic in many cases, as will be seen in Section 2.3).
2. Component commutability — free constructions allow their components to be substituted with other words, as long as selectional restrictions are not violated, whereas multiword lexemes usually do not, which is related to semantic compositionality. Since the components of the latter display “abnormal” meaning, they cannot be replaced with words that are their synonyms in regular situations.
3. Modifiability — free constructions can be arbitrarily modified (i.e. undergo morphosyntactic operations) to the extent permitted by general grammatical rules of the language. In the case of multiword lexemes, additional constraints apply, and sometimes no modifications are permitted at all.

2.2. Classification

As it was noted in the previous section, linguists often refer to multiword lexemes having different things in mind. Hence, it is important to precisely establish the scope of the notion and list the kinds of linguistic units that it covers.

This is not an easy task, however. If the only property that is required of multiword lexemes is conventionality, which is a continuum and not a set of discrete states, then it is clear that in some cases it will be very difficult, if possible at all, to tell whether a given unit is a multiword lexeme or not.

In cases such as *kick the bucket* judgement is obvious — it is definitely characterized by all the properties listed by Nunberg, Sag, and Wasow (1994), and therefore it is a multiword lexeme. But units characterized only by conventionality are not so straightforward to classify. Judgements regarding the degree to which a unit is conventionalized differ from speaker to speaker. *Hard drug* would probably be classified as conventionalized by most native speakers, but *bad weather* might be problematic (there are other adjectives that could be used to represent the same concept; this is not the case with *hard* in *hard drug*). Even the more general definition of Brundage et al. (1992) relies on human judgement whether such units are a part of the lexicon or not.

From the lexicographic point of view, the distinction here is between units that are conventionalized enough to be included in a dictionary as separate entries, and ones that are not. This will be discussed in more detail in Chapter 3.

⁵See Section 2.3 for a more detailed discussion of the last two points.

All these issues affect attempts at classifying multiword lexemes. Some researchers focus on highly conventionalized units, other ones include in their taxonomies even quite loose collocations. Each time it depends on the purpose of their work. The following subsections present two taxonomies, representing a theoretical and a practical approach to multiword lexemes.

2.2.1. Formal grammar approach

A taxonomy of multiword lexemes constructed from the point of view of formal linguistics can be found in Fillmore, Kay, and O'Connor 1988. The authors of the article were interested in fitting multiword lexemes into a generative grammar of English. In order to achieve it, they proposed their own set of properties and divided multiword lexemes into a set of categories.

Fillmore's team identified four sets with two properties each. The properties in all the sets form binary oppositions. First of all, a multiword lexeme can be either *encoding* or *decoding*. A decoding multiword lexeme is "an expression which the language users couldn't interpret with complete confidence if they hadn't learned it separately" (Fillmore, Kay, and O'Connor, 504-505). On the contrary, an encoding multiword lexeme is one whose understanding requires some special rules, but a speaker of the given language could nevertheless guess its meaning if he did not know them. The first group includes such units as *pull a fast one*, the second one — *wide awake*.

Secondly, there are *grammatical* and *extragrammatical* multiword lexemes. The former include units that follow familiar grammatical patterns and constructions, such as *spill the beans*. The latter are constructions which a general grammar of a language cannot successfully account for because outside of the multiword lexemes in question such constructions are considered to be ungrammatical. They include e.g. *by and large* (which is an anomalous coordination) and *at hand* (in which a determiner is missing).

Thirdly, Fillmore distinguishes *substantive* and *formal* multiword lexemes. Substantive units are fixed as far as their lexical content is concerned, nothing can be added or subtracted from them. These include e.g. all the units listed in the previous paragraph. Formal multiword lexemes are much more interesting because they are not rigidly fixed and function rather as syntactic patterns, and not concrete phrases.

A very significant point raised by Fillmore concerning formal multiword lexemes is that often they act as hosts for substantive ones. The substantive multiword lexeme in (1) is a special instance of a more general pattern shown in (2):

- (1) *The bigger they come, the harder they fall.*
- (2) *the faster you do it, the less accurate it will get
the ..., the ...*

Lastly, there are multiword idioms that have a special pragmatic purpose and those that do not. The first group includes many substantive units (*good morning*, *how do you do*), but formal ones are represented in it as well (*Him be a doctor?*⁶).

On the basis of the property sets listed above Fillmore builds a typology of multiword lexemes which consists of three categories and revolves around the concept of familiarity which will be discussed in much more detail in Section 2.3.

⁶In Fillmore's opinion such syntactic patterns with little fixed content are multiword lexemes.

The first category is labeled *unfamiliar pieces unfamiliarly arranged*⁷. Substantive multiword lexemes that belong to this category are all the ones that are composed of, or contain, obsolete words which appear only in the unit in question (e.g. *spick and span*).

The second category is called *familiar pieces unfamiliarly arranged*. Substantive idioms which belong to it include e.g. *all of a sudden* and *in point of fact*. Among formal idioms of this kind are expressions used for constructing kinship terms, such as *second cousin three times removed* or *first cousin twice removed*. Their unfamiliar arrangement becomes obvious if one considers the following phrases:

(3) ?*fourth chapter three times rewritten*

(4) ?*second book twice published*

In Fillmore's opinion, accounting for such expressions requires specialized mini-grammars to be embedded in general grammars, whose properties cannot be deduced from the latter. It is interesting to notice that the idea of using small local grammars for the treatment of multiword lexemes became significant in 1990s, when the IDAREX formalism was created at Xerox Research Center Europe (see Chapter 4).

The last category distinguished by Fillmore is *familiar pieces familiarly arranged*. These are expressions built of common lexical units according to common principles, but with idiomatic interpretations assigned. They include such units as *pull someone's leg*, and rhetorical questions that communicate negative emotions: *Who's gonna make me?*

2.2.2. NLP approach

A very interesting approach to classifying multiword lexemes can be found in the already mentioned work by computer scientists from IBM (Brundage et al. 1992), which was prepared for the purposes of a machine translation project. Instead of listing abstract categories with hand-picked examples, the authors decided to use a low-level approach and classify multiword lexemes according to their syntactic structure on the basis of a set of 300 English and German phrases.

The list of structures presented in the report is very broad. The following is just a sample which is supposed to give an idea how the classification was done (only the set of top-level categories is complete):

1. Verbal multiword lexemes

(a) PP + VP

i. Prep. + (Adj.) + N + VP
vor Wut kochen

ii. Prep. + Def. Art. + (Adj.) + N + VP
am Ball bleiben

iii. ...

(b) NP + VP

⁷The authors emphasize that "in describing pieces as unfamiliar we must recognize that they are not all completely unfamiliar" (Fillmore, Kay, and O'Connor 1988, 508).

- i. N + N + VP
Blut und Wasser schwitzen
 - ii. ...
- (c) ...
- 2. Adjectival multiword lexemes
 - (a) Adv. + Participle/Adj.
highly improbable
 - (b) ...
- 3. Noun multiword lexemes
 - (a) N + Genitive Attr.
the devil's advocate
 - (b) ...
- 4. Other multiword lexemes
 - (a) Adverbial multiword lexemes
now and then
 - (b) Function word multiword lexemes
 - i. Prepositions
on behalf of
 - ii. ...

The strength of this approach is that no categories are ambiguous and that one can see what really is contained in them — formal syntactic patterns turn out to be much more legible than lengthy descriptions. The downside is that the study cannot represent the whole lexical stock adequately, as it was based on just 300 phrases. Also, due to the nature of the approach, it is far from being universal and can only claim to describe German and English, and even them only to some extent.

2.3. Lexical and syntactic variability of multiword lexemes

The most interesting property of multiword lexemes from among all the ones mentioned in Section 2.1 is their variability. In many cases it makes them very similar to free combinations that do not have a lexemic status, and thus leads to numerous problems and difficulties. In the case of lexicography, the problems can be divided roughly into two groups:

1. From the lexicographer's point of view, the problem is with deciding which multiword lexemes are "frozen" enough to be included in a dictionary, and which ones are so flexible that including them would be impractical (at least in a printed dictionary).

2. From the dictionary user’s point of view, the difficulty is with identifying that an expression in a text is a multiword lexeme and with looking it up — the more flexible a multiword lexeme is, the harder it becomes to establish its base form and look it up in a dictionary effectively.

Multiword lexemes’ variability can be roughly divided into two kinds — lexical, in which some of the words that constitute a unit may be added, deleted, or substituted with other ones (chosen from a strictly limited or an open set), and syntactic, which allows various syntactic transformations to be performed upon the units in question.

Both kinds of variability give birth to many “surface” realizations of the same “underlying” multiword lexemes, which in many cases are much too numerous to be listed in a dictionary, but nevertheless require lexicographic description if dictionaries are to provide adequate information for language learners and translators. A successful and exhaustive description of such modifiable units requires computational tools which will be described in more detail in Chapter 4.

The purpose of this section is to present various theoretical explanations for the fact that multiword lexemes are syntactically and lexically flexible. Firstly, it will demonstrate the extent to which such units are variable and the transformations they may undergo. Secondly, it will present approaches to variability based on *semantic compositionality* and ones that revolve around the notions of *age* and *familiarity*.

Most authors usually mix the two types of variability distinguished here (*lexical* and *syntactic*). Moreover, often the same transformations are given different names by individual authors or are combined into broader groups. What follows is a compilation of the most important transformations that multiword lexemes may undergo. The examples listed here are also to demonstrate that individual modifications are allowed in the case of some multiword lexemes, but lead to ungrammaticality or loss of idiomatic meaning in others, which will become more important in Section 2.3.4.

2.3.1. Lexical operations

Lexical operations function on the level of individual words. They constitute the set of “weak” transformations that some multiword lexemes might undergo⁸. The following set is based on Guenthner and Blanco 2004, in addition to Nunberg, Sag, and Wasow 1994.

Modification — insertion of additional modifying words or phrases (adjectives, adverbs, subordinate clauses, etc.):

- (1) *kick the filthy habit*
- (2) *your remark touched a nerve that I didn’t even know existed*
- (3) *a very *black hole*
- (4) **kick the bucket hard*

An operation which could probably be considered to be a subset of modification is *quantification*:

⁸In all the subsequent examples asterisks indicate that the marked phrase lost its idiomatic meaning due to the transformation. Question marks are used to signal units whose status in that respect is problematic.

- (5) *we could pull yet more strings*
- (6) **shoot another breeze*

Substitution of synonyms — replacement of a word or words within a multiword lexeme with another one that has similar meaning:

- (7) *an iron hand/fist in a velvet glove*
- (8) *a white dwarf/*gnome*

Deletion — removal of optional lexical units:

- (9) *an iron hand* (in a velvet glove)

2.3.2. Syntactic transformations

The second set of operations that multiword lexemes may be subjected to are syntactic transformations which operate on the phrasal level and usually involve complex restructuring of the linguistic units they are applied to. The following set has been compiled on the basis of works by Nunberg, Sag, and Wasow (1994), Guenther and Blanco (2004), and Reagan (1987).

Inflection — ubiquitous transformations that involve alterations of tense, number, gender, etc.:

- (10) *John was on the verge of jumping.*
- (11) *Harry kicked the bucket yesterday.*

Nominalization — transformation of a verb phrase into a noun:

- (12) *?their spilling of the beans*
- (13) **his sawing of the logs*

Topicalization — emphasis placed on the topic or focus of a sentence by preposing it to the beginning of the sentence:

- (14) *His closets, you might find skeletons in.*
- (15) **The bucket, he kicked two days ago.*

Ellipsis — omission of a word or words necessary for a construction to be complete, but understood in context:

- (16) *My goose is cooked, but yours isn't.*
- (17) **John saws logs, and George doesn't saw them.*

Particle movement — moving the particle of a phrasal verb away from its head verb:

- (18) *The charity passed the hat round.*

Passivization — transformation of an active sentence into the passive voice:

- (19) *The beans were spilled.*
- (20) **The logs were sawed.*

Raising to subject — movement of the subject of a sentential complement to the subject position of the whole clause:

(21) ?*The hat is likely to be passed around.*

(22) **The bucket is likely to be kicked.*

Clefting — forming a redundant complex sentence for the purposes of emphasis:

(23) ?*It were the beans that they spelt.*

(24) **It were the logs that he sawed.*

All the transformations listed in linguistic literature form a rather chaotic set which some linguists tried to order. Fraser (1970) distinguished six classes of syntactic transformations and ranked them from the least to the most disruptive from the point of view of multiword lexemes' semantic properties — the most disruptive transformations lead to the loss of idiomatic meaning. The latter end of the scale was occupied by topicalization, which Fraser believed to be impossible to impose on any multiword lexeme while retaining its idiomatic meaning⁹. Fraser did not try to explain flexibility however, and did not use real-life linguistic data — he conducted his research by means of introspection. Nevertheless, other linguists analyzed his results later on in a more objective fashion and found them to be mostly correct (see Section 2.3.4).

2.3.3. Internal and external transformations

Transformations of multiword lexemes can be divided into *internal* and *external* (Brundage et al. 1992). However, such a classification would be hard to apply to the above sets.

Internal modifications do not affect the meaning of multiword lexemes, e.g. the noun in *an iron hand/fist* can be chosen from a set of two without any influence on the unit's meaning. On the contrary, external modifications change the semantic properties of multiword lexemes, e.g. *to kick the habit* can have adjectival modifiers added to it, such as *dirty* or *disgusting*, which contribute additional meaning to the phrase.

The reason why it would be hard to classify the above set of transformations according to whether they are internal or external is that some of them are both, depending on the context. Modification is a good example — it can be either external and affect the meaning, as in *kick the filthy habit*, or internal and therefore neutral, as in *thick as two (short) planks*, in which *short* is optional and does not add up to the phrase's meaning.

2.3.4. Explaining multiword lexeme variability

As it has been shown in the examples quoted in the previous section, multiword lexemes differ in the respect of their ability to undergo transformations. An obvious question one might raise here is why some units are easier to modify than others. The following sections present two attempts at answering this question — one is based on the notion of *semantic compositionality* and the other one on *age* and *familiarity*.

⁹Which does not seem to be correct, as can be seen in the examples above.

Semantic compositionality

A detailed explanation for multiword lexeme flexibility in terms of semantic compositionality can be found in Nunberg, Sag, and Wasow 1994. The authors' first important claim is that the traditional definition of multiword lexemes which is based on their alleged non-compositionality (see Section 2.1) is grounded on a misconception, because actually the meaning of most of them can to some extent be derived from the meaning of their constituents. In order to justify the non-compositionality claim, it is not enough to prove that a multiword lexeme's meaning cannot be predicted on the basis of the meaning of its parts. It also has to be shown that "once the meaning of the idiom is known (say by hearing it used in a sufficiently informative context), it cannot be devolved on the constituents of the expression. And this is not entailed by simple nonpredictability" (Nunberg, Sag, and Wasow, 496).

To illustrate it, the following example is given:

- (26) *John was able to pull strings to get the job, since he had a lot of contacts in the industry.*

Even though one would most likely be unable to conclude that *pull strings* means 'take advantage of connections' if he heard the phrase in isolation for the first time, the context allows him to establish the meaning correctly¹⁰. This in turn makes it possible to try to relate the concepts denoted by the phrase (taking advantage, connections) to its constituents (*pull*, *strings*). Such an analysis leads to the conclusion that individual constituents metaphorically refer to the parts of the interpretation. In the authors' opinion, this will give the multiword lexeme "a compositional, albeit idiosyncratic, analysis" (Nunberg, Sag, and Wasow, 496).

This might look like abandoning the idea of conventionality (cf. Section 2.1) of multiword lexemes, but it is not so. The argument does not mean that no conventions are involved. It means that conventionality is rather to be sought in the individual constituents, and not the whole multiword lexeme. In the authors' opinion, it is conventions that allow speakers to metaphorically refer to connections by means of *strings*, and to exploiting them by means of *pulling*. Saying that these relations are conventionalized allows the authors to disregard the question of why a particular word is used to denote a particular concept.

However, not all multiword lexemes allow a compositional analysis. There are ones such as *saw logs* or *shoot the breeze* in the case of which it is impossible to relate the parts of their interpretation to their constituents. This distinction led the authors to divide multiword lexemes into two separate types: *idiomatically combining expressions* which can undergo compositional analysis, and *idiomatic phrases* which cannot (cf. Figure 2.1 on the following page).

All this has important implications for the variability of multiword lexemes, because according to the authors modifications and transformations are imposed not on the whole units, but rather their parts. In order for these transformations to produce grammatical results, the parts (or chunks) in question need to carry identifiable meaning when used idiomatically.

The authors list the following operations that idiomatically combining expressions may be subjected to (cf. Sections 2.3.1 and 2.3.2):

¹⁰The authors acknowledge that compositionality in such cases is rather weak because context is required to work out the meaning (Nunberg, Sag, and Wasow 1994, 499).

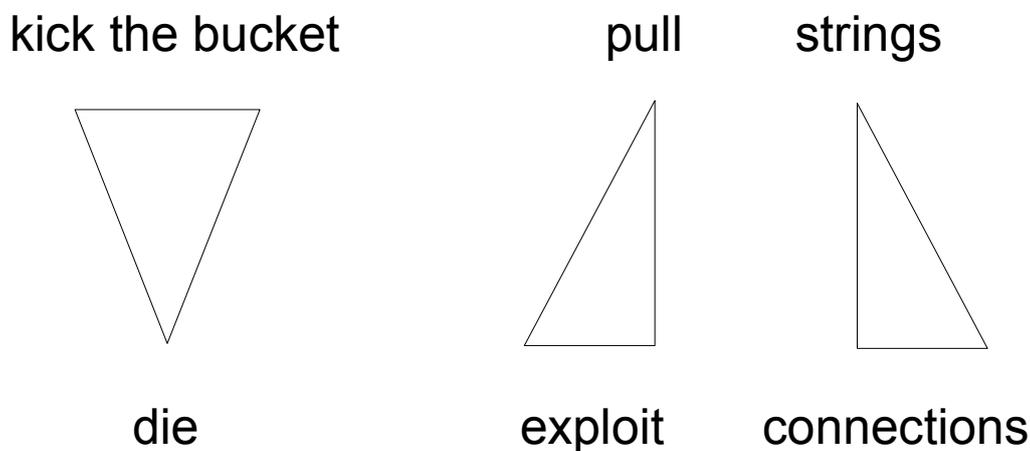


Figure 2.1: Compositionality of meaning in idiomatic phrases and idiomatically combining expressions.

- passivization/raising — *advantage seems to have been taken of Pat*
- quantification — *take no significant advantage*
- ellipsis — *they claimed full advantage had been taken of the situation, but none was*

Idiomatic phrases, on the contrary, cannot undergo any such operations without having their semantic properties altered. When transformed, such phrases always lose their idiomatic interpretations (*the bucket was kicked by him* \neq *he died*).

Therefore, the only multiword lexemes that can undergo lexical and syntactic operations are idiomatically combining expressions. This in turn implies that semantic compositionality (and, by extension, meaning) is responsible for the varying degree to which multiword lexemes are variable¹¹.

A similar argument is repeated by Cruse (1986, 38): “the reason that *to pull someone’s left leg* and *to kick the large bucket* have no normal idiomatic interpretation is that *leg* and *bucket* carry no meaning in the idiom, so there is nothing for *left* and *large* to carry out their normal modifying functions on . . .”.

The partial compositionality claim is also supported by psycholinguistic research. Language users process simple lexical units that do not have any internal structure rapidly. Since idiomatically combining expressions are partly compositional, one would

¹¹At least in English — the authors quote examples from German in which syntactic variability does not require semantic analyzability. However, the authors claim that German transformations, although similar to their English counterparts on the surface, are underlyingly different operations (Nunberg, Sag, and Wasow 1994, 512-513).

expect them to be processed less quickly than idiomatic phrases (because they are not uniform wholes, but rather collections of smaller parts, and therefore some structural analysis is necessary). Reagan (1987, 421) quotes research results which prove that such a difference in processing speed exists. Therefore, theoretically one should be able to predict how flexible a multiword lexeme is by measuring the time in which it is processed by language users.

Age and familiarity

A study of how the age of multiword lexemes is related to their flexibility can be found in Cutler 1982. The author selected a sample of idioms from Fraser 1970 and checked their age against the Oxford English Dictionary. Her results showed that “frozenness and age are not perfectly correlated; but there is a reliable tendency for the more frozen idioms to have been longer in the language” (Cutler, 318). This in turn leads to two hypotheses about the units’ flexibility.

One is that becoming inflexible is a gradual process that takes place over decades or centuries. In order to check it, one would need to consult early sources and see whether multiword lexemes that are frozen nowadays were used in a more flexible manner before. The other hypothesis says that multiword lexemes become inflexible when their meaning is no longer obvious, and their literal reference has become forgotten. In many respects this is similar to what has been presented in Nunberg, Sag, and Wasow 1994. Inability to decompose the interpretation of a multiword lexeme and relate it to its parts may result from the fact that speakers have forgotten e.g. what real-life object is denoted by *bucket* in *kick the bucket*¹².

Another approach to explaining the flexibility of idioms is presented in Reagan 1987. The author ran a series of five experiments on a group of Harvard undergraduates. Each experiment consisted of four tasks:

- judging acceptability of multiword lexemes subjected to various transformations (from Fraser 1970, with several additional ones¹³)
- providing definitions for multiword lexemes (i.e. measuring how “familiar” the subjects were with them)
- judging “meaning closeness”, i.e. comparing the idiomatic meaning of multiword lexemes to what they would mean should their constituents be interpreted literally
- judging “explicability”, i.e. trying to provide etymologies for multiword lexemes’ idiomatic meanings

Reagan calculated statistical correlations between the flexibility of multiword lexemes and their familiarity, meaning closeness, and explicability. Familiarity turned out to be the most strongly correlated with flexibility — “the best predictor of how flexible an idiom is appears to be the fraction of the population who know its meaning” (Reagan, 435).

¹²*Bucket* refers to a hook on which slain animals were hanged by their legs in abattoirs — hence the verb *kick*.

¹³Reagan’s experiments showed that Fraser’s individual intuition about the ordering of transformations was mostly compatible with what a randomly selected population of subjects said.

This seems to be inconsistent with the analyses based on age and semantic compositionality. E.g., *kick the bucket* is familiar to most of the population, but nevertheless it is one of the least flexible multiword lexemes, probably because of its long history and the fact that its literal reference is no longer remembered. It remains to be measured which of the three views provides the biggest number of correct judgements concerning the potential of multiword lexemes to undergo transformations.

2.3.5. A practical approach to describing variability

The research results presented above are valuable, but they focus on the theoretical side of the matter and rely on very limited sets of data. As such, they are not very useful for practical applications which require information about the *specific* transformations that *specific* multiword lexemes can undergo.

For such information one should once again turn to the IBM report (Brundage et al. 1992). Apart from compiling a detailed list of multiword lexeme structures (see Section 2.2.2), they also prepared a set of low-level (compared to the rather general ones listed above) modifications which they can undergo. The set includes number variation, determiner modification, comparative form, adjectival modification, adverbial modification, passivization, negation, coordination, referential potential¹⁴, and word order variation.

The authors subjected each multiword lexeme from their corpus to all the relevant transformations and checked whether the resulting phrases were grammatical and whether they retained their idiomatic meaning. The report contains detailed tables in which it is clearly visible which operations can be applied to individual units. It is probably impossible to come up with a more detailed and unambiguous study, but the drawback mentioned in Section 2.2.2 is still valid — the study is limited to 300 phrases. Nevertheless, performing such a study for a bigger set of linguistic units is probably necessary for any system that is supposed to generate natural language correctly¹⁵.

¹⁴I.e. anaphoric references, forming questions, and relative clauses.

¹⁵Less so for systems that are only supposed to accept linguistic input, as their designers can assume that there is no “wrong” input, disregard the restrictions, and allow all modifications of multiword lexemes — see Chapter 4.

Chapter 3

Multiword lexemes in printed dictionaries

As it has been shown in the previous chapter, the phenomenon of multiword lexemes is very complex. The sheer number of such units, the troubles with classifying them, and the fact that they are lexically and syntactically flexible make any attempts at providing an adequate lexicographic description of them very difficult. One could even argue that none of the available printed dictionaries provides a formalized and accurate description of multiword lexemes. Such claims are made e.g. by the members of the XMELLT project¹: “...it is widely acknowledged that current printed dictionaries do not contain information about multi-word expressions in a coherent and exhaustive way...”.

Yet giving at least a partial account of multiword lexemes is necessary for any dictionary that claims to adequately describe a language or provide an interface between two languages. According to the XMELLT project, “multi-word constructions are extremely frequent in language, comprising perhaps 30% of the lexical stock...”. No adequate monolingual dictionary can ignore such a common phenomenon, bearing in mind that “it is [the lexicographer’s] duty to find out and describe the lexical units of the language” (Zgusta 1971).

As far as translation dictionaries go, there is general agreement among professional translators and researchers that translation takes place above the level of words (Grosbart 1987). Multiword lexemes seem to be a good approximation of the level on which translation actually takes place, and therefore their incorporation into dictionaries aimed at translators is absolutely necessary. Because of all these facts, most printed dictionaries, both mono- and multilingual, have tried to account for multiword lexemes and to describe them in the most formal and coherent fashion possible, despite the problems posed by their number and flexibility.

The current chapter summarizes the lexicographic approach to multiword lexemes and the guidelines for incorporating them in printed dictionaries. It also looks at several dictionaries to see how these guidelines are realized in practice.

¹Cross-lingual Multi-word Expression Lexicons for Language Technology. <http://www.cs.vassar.edu/~ide/XMELLT.html> Accessed on April 25, 2006.

3.1. Selection

Up to this point the current thesis has been concerned with multiword lexemes in the broadest possible meaning of the term, covering all units ranging from compound nouns to syntactic patterns with little fixed content. Since this chapter deals with lexicography, the scope of the notion should be slightly narrowed because “. . . the lexicographer is not primarily interested in whole sentence-patterns, the study of which belongs rather to syntax” (Zgusta 1971, 138). Lexicographers are concerned with “concrete” units which can be put into dictionaries, either as headwords or subsenses of one of their constituents. More abstract phenomena, such as *the faster*, *the better* and *inch by inch* patterns, in which all meaningful constituents are variable, are not a subject of lexicography in the traditional sense, partly because such units could not be easily integrated into the structure of traditional general-purpose dictionaries whose macrostructure is based on alphabetical order.

When listing types of multiword lexemes for the purpose of providing guidelines for lexicographers, Benson, Benson, and Ilson (1986) look at the whole problem from the point of view of flexibility and meaning compositionality, which were discussed in Chapter 2. On one end of the spectrum they distinguish *free combinations*, i.e. clusters of words created for the purpose of immediate utterances, whose meaning can be predicted from the meaning of their constituents. On the other end there are idioms which allow for little or no lexical and syntactic variation, and whose meaning is (in the authors’ opinion) non-compositional.

Between the extremes Benson, Benson, and Ilson list *collocations* (compositional meaning, strong ties between words and hence low flexibility), *transitional combinations* (more frozen than collocations, meaning *close* to compositional), and lastly *compounds* (completely frozen and semantically compositional).

As far as which multiword lexemes should be included in dictionaries is concerned, the authors say that:

Free combinations should ordinarily be included in dictionaries *only* when they are needed to exemplify the meaning of a word, especially if it is polysemous. . . . On the other hand, the compiler should include as many idioms, collocations, transitional combinations, and compounds as possible. The choice of the items to be entered depends on the planned size of the dictionary and on the skill of the compiler in selecting those combinations that are most vital to the dictionary description of English. (Benson, Benson, and Ilson 1986, 254-255)

It follows that the process of selecting units for inclusion in a dictionary consists of two stages. Firstly, there is the initial selection during which lexicographers decide upon the *types* of multiword lexemes that are supposed to be included. Secondly, they choose specifically *which* individual lexemes will make it into the dictionary, which is determined primarily by practical factors — the planned size of the dictionary and the compilers’ abilities.

Zgusta (1971) divides multiword units into *free combinations* and *set combinations*. The former are defined similarly to what can be found in Benson’s work. The author does not postulate including them in dictionaries, but he gives several reasons why they are useful for lexicographers. Firstly, they show the most typical usage patterns of a word, which can be illustrated in a dictionary in examples and thus provide valuable information e.g. for second language learners. Secondly, by studying free combinations

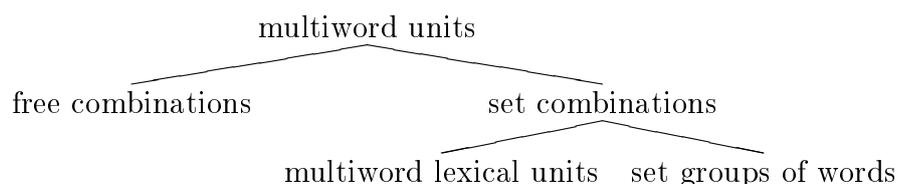


Figure 3.1: Zgusta’s classification of multiword units.

lexicographers can determine whether there are restrictions in a word’s combinatorial potential. Moreover, often two words that are basically lexical equivalents in two languages have very different combinatorial properties (consider English *eat*, and Polish *jeść* and *zreć*). Free combinations also allow lexicographers to notice multiple meanings of a single word (*cut some string* vs. *tokenize a string*).

Among set combinations Zgusta distinguishes units that should be included in dictionaries as separate entries, and ones that should not. The former are *multiword lexical units* (which basically are a subset of multiword lexemes in this thesis’s understanding of the term). The latter include such phenomena as proverbs, sayings, dicta, and famous quotations (“to be or not to be”), which Zgusta collectively labels *set groups of words*. A diagram of Zgusta’s classification is shown in Figure 3.1.

Contrary to Benson, Benson, and Ilson (1986), Zgusta gives precise guidelines that say which multiword lexemes should be left out if space is scarce. The criterion according to which the phrases should be selected is their relative *setness*, which means that the more conventionalized and frequently used a combination of words is, the more worthy it is of being included in a dictionary: “the smaller the dictionary, the more severe will be [the lexicographer’s] choice of examples in favor of the most stabilized set combinations: the bigger the dictionary, the greater the opportunity to begin with less setness” (Zgusta 1971, 155).

3.2. Identification

Another problem lexicographers need to face before the actual compilation of a dictionary is identification of the concrete members of the abstract types defined in the stage of selection. Benson et al. propose only one test for identifying multiword lexemes (and actually, only a particular subset of them): in order to see whether the unit in question is a compound, one need to check whether it has a one-word counterpart in other languages — if this is the case, there is a good chance that it is a compound which should be listed as a separate entry².

Zgusta’s manual contains a much broader and comprehensive set of tests by means of which one can quite reliably tell whether a combination of words really is a conventionalized multiword lexeme that acts as a member of the lexicon. The set includes the following:

- Multiword lexical units, and set combinations in general, do not allow substitution. For example, one cannot say *Excellent morning!* instead of *Good morning!*

²Such tests should become easier to perform with the emergence of electronic, crosslingual lexicons of multiword lexemes — see Villavicencio, Baldwin, and Waldron 2004.

when greeting somebody. Substitution is only possible with free combinations³.

- Not all multiword lexical units permit addition. For example, it is not correct to say *black steel market* — one can only say *black market in steel* (although *illegal steel market* is fine, but *illegal market* is a free combination).
- Many multiword lexical units are characterized by non-compositional meaning. This is most frequently the case with *set expressions* which roughly correspond to *idioms* in the sense of Nunberg, Sag, and Wasow 1994.
- The usage of a set combination's constituent may be strictly limited to this very combination, as it is in the case of *fro* in *to and fro*.
- Quite often multiword lexical units have one-word synonyms, e.g. *guinea pig* :: *cavy*.
- Multiword lexical units are also often characterized by having a one-word counterpart in foreign languages, e.g. French *pomme de terre* and English *potato*.
- Another peculiarity of multiword lexical units is that sometimes they exhibit special grammatical properties. This is exemplified by such phrases as *at hand* and *by heart* in which nouns do not take any articles.

When looking for multiword lexical units, one has to watch out for phrases that Zgusta labels *stereotyped expressions* or *clichés*. They include stock phrases that are only too frequent in bad prose, low-profile newspapers, and the vernacular of politicians or sport journalists. As far as Polish is concerned, one should think of such combinations of words as *sprawa o podłożu politycznym* or *państwo prawa*. Because of extremely frequent usage, such phrases have become semantically empty. In Zgusta's opinion they are not multiword lexical units, but merely very common free combinations.

Another difficulty is highly specialized, technical, and scientific vocabulary. In the jargon of computer science, *hash table* and *routing table* are most probably treated as multiword lexical units. However, to the general public, those are nothing more than different specifications of a table. Lexicographic treatment of such units should probably be based on the level of specialization of the planned dictionary.

Zgusta does not describe specific techniques of looking for multiword lexemes, but since his book comes from 1971, one can safely assume that he had manual methods in mind. However, the scale of the phenomenon implies that in order to be successful, any attempts at identifying a representative set of multiword lexemes for a language should be aided by automatic methods that make it possible to extract such units from language corpora⁴. This is especially important in the case of collocations which can be reliably identified only by means of statistical analysis (nowadays this can be achieved with free⁵ software — see <http://www.mimuw.edu.pl/polszczyzna/kolokacje> and Buczyński 2004).

³As it has been shown in Section 2.3, certain units that definitely deserve to be included in dictionaries allow substitution of synonyms — e.g. *to kick/cool one's heels*.

⁴This goes much beyond the scope of this thesis. However, information on this topic is abundant; a simple Google query of the type `multiword +expression +extraction` returns 13,200 hits.

⁵In the sense of the Free Software Foundation. See <http://www.fsf.org>

At one point Zgusta acknowledges that the frequency of co-occurrence can be seen as another test for identifying multiword lexical units. However, he also says the following: “I do not know whether this is as useful a criterion as it would seem in our statistical age. I do not know of any conclusive count which could give us some undubitable examples” (Zgusta 1971, 151). Zgusta recommends taking results of statistical research with a grain of salt — an advice that is still valid, although nowadays collocations are identified by means of sophisticated statistical tests tweaked for linguistic analysis. In Zgusta’s words:

...I strongly suspect that the frequency of the co-occurrence of two words may be even greater if we have to do with a fully free combination of words which have themselves a high frequency of occurrences; e.g. a statistical count would probably show that the combination *to drink beer* has an immensely higher frequency of occurrence than *to swallow stones*, but both are free combinations anyhow. (Zgusta, 151)

Identification of multiword lexemes is not only a lexicographers’ problem. Also second language learners have difficulties with identifying such units in texts⁶. Obviously, units that cannot be identified also cannot be successfully looked up in a dictionary. In such cases computer-aided *context-sensitive dictionary lookup* is probably the only useful solution (see Chapter 4).

3.3. Lexicographic practice

Keeping in mind the number of existing multiword lexemes and the size restrictions of even the biggest dictionaries, in addition to the fact that the selection of multiword lexemes to be included in a dictionary is determined by subjective judgements of dictionary compilers, it comes as no surprise that lexicographers treat such units in a highly inconsistent way. The following sections present the ways in which multiword lexemes have been accounted for by general-purpose and specialized dictionaries.

3.3.1. General-purpose dictionaries

A detailed study of how multiword lexemes are treated in general-purpose dictionaries can be found in Gates 1988. The author’s analysis includes six major lexicographic works: The American Heritage Dictionary of the English Language (Second College Edition), Chambers 20th Century Dictionary (New Edition), Collins English Dictionary, Longman Dictionary of the English Language, Webster’s Ninth New Collegiate Dictionary, and Webster’s New World Dictionary (Second College Edition). It is focused on four main areas: inclusion of multiword lexemes, their place of entry, the forms of the lemmata, and the additional information that was included.

Inclusion

For every listed dictionary, Gates took a sample composed of the first 500 entries under the letter “R” in order to calculate proportions between regular words and multiword

⁶This also concerns transparent multiword lexemes which have been “obfuscated” by lexical and/or syntactic transformations.

lexemes. The results showed that the extreme cases were Chambers, which had 33.5% of multiword lexemes in the sample⁷, and Webster's Collegiate which had only 16%.

Gates's general conclusion is that dictionaries do not include customary strings of words that do not exhibit any idiomaticity in their meaning or grammatical properties. Despite this fact, they often do include *transparent collocations*, such as names of objects (e.g. *rocking chair*) or specialized terminology (e.g. *immediate constituent*). Inclusion of phrases that do exhibit idiomaticity or are anomalous from the point of view of grammar is not consistent. For example, *in the know* (which has a verb as the object of the preposition) is generally included, but *come what may* was not present in any of the tested dictionaries.

Inconsistencies are also prominent in the treatment of polite forms like *thank you* (included only in two of the tested dictionaries) and *how do you do*. Interjections such as *that is to say* are also often omitted (probably due to their familiarity which partly obscures the fact that they are anomalous).

In general, lexicographers are reluctant to include frequently occurring clauses, sentences, and other independent combinations which exhibit a certain degree of conventionality. The author claims to have identified over a hundred units like *go fly a kite* and *Has the cat got your tongue?* which are not adequately covered by any of the dictionaries he examined. Proverbs, which constitute a subclass of sentences, are also generally not present in general-purpose dictionaries, mainly because there are specialized works that deal with them exclusively.

Dictionaries commonly include multiword lexemes composed of words unique to the unit in question (*spick and span*) and multiword lexemes whose constituents are not used in any other context (*kith and kin*). Special constructions made up of pairs of words (*either...or*, *as...as*) are usually included, however not as separate entries, but rather subsenses of the words.

The last class of multiword lexemes that Gates analyzes are patterns with little fixed lexical content, such as *inch by inch*, *more and more* and *hour after hour*. These are said to be very problematic for lexicographers, but nevertheless the dictionaries that were examined include inconsistent numbers of the members of several of these sets.

Place of entry

As far as the place of entry of multiword lexemes is concerned, 5 out of 6 analyzed dictionaries included some of them as main entries. Those were compound nouns, noun phrases like *rule of thumb*, hyphenated verbs like *rubber-stamp*, and foreign phrases like *ad hoc*. Other types of multiword lexemes are treated in inconsistent ways. Most of the dictionaries include compound conjunctions as main entries. One of them does that for phrasal verbs.

There are several problems with placing multiword lexemes in subentries. First and foremost, lexicographers need to decide in the entry of which component should the whole unit appear. Usually this is the first major invariable word, but detailed policies differ from one dictionary to another. Besides, there is much trouble with multiword lexemes whose first content word is unique to the combination, such as in *Achilles heel*.

The problems are not limited to dictionary macro-structure. Lexicographers also differ in regard to the exact position within an entry at which multiword lexemes should

⁷This roughly corresponds to the estimates of the XMELLT project.

be placed if they are treated as subentries. However, this does not seem to be a very significant issue unless it leads to inconsistencies within a single dictionary.

Lemmata

A substantial amount of problems that lexicographers have with multiword lexemes results from their flexibility. Specifically, it is not always clear what lemma is to be used if a multiword lexeme's constituents are variable. For example, the expression *chew the rag* also exists in the form *chew the fat*. Technical means of handling such cases in dictionaries are inconsistent. Some of the dictionaries examined by Gates list one variant after the other: *chew the rag (or fat)*. Others print the phrase twice with different wording, or include a note at the end of the entry (in this case it might say: *also: chew the fat*). When variants are very numerous, lexicographers either use the phrase *etc.* — as in *up to one's ears etc.*, or try to list as many of them as possible — *up to one's ears/eyes/armpits/neck*.

Optional words in multiword lexemes are often parenthesized. In some cases their presence is signalled by a note, e.g. in the case of *over and over* it might say: *often followed by* again. The last problem is variable possessors and personal objects. The former are usually marked with *one's* (when the possessor refers to the sentence's subject, e.g. *make up one's mind*) or *somebody's* (when it does not, e.g. *break somebody's heart*). Variable personal objects are usually marked with *someone*.

Additional information

The last area that Gates analyzes is the information which multiword lexemes are accompanied by. It seems that providing information about pronunciation depends on orthography. None of the studied dictionaries provides pronunciation of compounds that are main entries written without hyphenation. Most of the dictionaries give pronunciation of hyphenated words; some of them provide only the stress patterns (in the case of subentries). None of them provide stress patterns for all compounds, which is a pity since this is what second language learners have much trouble with.

Part-of-speech information is inconsistent across the dictionaries. None of them labels all multiword lexemes. Apart from that, policies differ: some dictionaries label all main entries, others only compounds, but not phrases like *bed of roses*. Collins English Dictionary sometimes provides usage-related information, e.g. *like hell* is labeled with *(adv.)(intensifier)*.

All the dictionaries feature definitions for multiword lexemes that have figurative meaning. This is especially true about units whose real meaning is very distant from their literal interpretation, e.g. *go to seed*. However, even more transparent units are often defined, as it is in the case of *get back*, whose meaning seems to be rather obvious even for non-native speakers of English.

Dictionaries do not provide etymological information for multiword lexemes whose origin is obvious. Etymologies are given usually only for words that come from foreign languages or the ones that contain words which are unique to a particular multiword lexeme, such as *runcible* in *runcible spoon*. However, not everything that seems obvious really is so. For example, *upside down* is derived from an earlier phrase, *up so down*. Etymological information is sometimes contained within definitions, e.g. Chambers says that *know the ropes* means “to understand the detail or procedure, as the sailor does his rigging”.

Suggestions

In the end Gates provides several suggestions for improving the treatment of multiword lexemes in general-purpose dictionaries. His most interesting points are the following:

1. Formulate policies of collection and selection that will include more multiword lexemes of value to dictionary users. Fewer transparent collocations and more idiomatic sentences are in order.
- ...
3. Deal consistently with quasi-lexemes like *day after day*.
4. Formulate simple policies regarding place of entry and explain them in the front matter of the dictionary.
- ...
8. Eliminate unnecessary literal senses.
- ...

(Gates 1988, 105-106)

3.3.2. Specialized dictionaries

Longman Dictionary of English Idioms

The “Longman Dictionary of English Idioms” (Long 1979), from now on LDOEI, was most probably the first printed lexicographic work that attempted to treat multiword lexemes in a highly formalized manner, which included creating a mini-language for dealing with their lexical and syntactic flexibility. The dictionary’s strengths also lie in the fact that it covers a broad range of such units, including ones that are hardly ever found in general-purpose dictionaries. In addition to a regular set of traditional idioms and sayings, the work also lists e.g. allusions (*Catch 22*) and typical conversational phrases (*how do you do?, so to speak, now you’re talking!*). Especially the latter seem to have been neglected by traditional dictionaries. A full list of the types of multiword lexemes described in the dictionary can be found in its introduction.

LDOEI’s macrostructure is based on alphabetical order. Multiword lexemes are listed under headwords that reflect their main constituents, thus *spill the beans* can be found under *beans* and is also cross-referenced under the headword *spill*. In addition to definitions, the dictionary also provides register labels, usage and grammatical information, examples, and historical explanations.

The most interesting thing about LDOEI is the formal language used to describe multiword lexemes. It consists of a set of operators listed below:

- $^{\circ}$ — indicates a word that may inflect or change form (*spill^o the beans*)
- \textcircled{D} — a wildcard symbol for the direct object (*get \textcircled{D} off one’s chest*)
- \textcircled{I} — a wildcard symbol for the indirect object (*give \textcircled{I} the first refusal*)
- \textcircled{P} — a wildcard symbol for the object of a preposition (*set foot on \textcircled{P}*)
- m — indicates that a word may move and can be placed either before or after the object (*pass the hat round^m*)

- x/y — indicates that word *y* may be used instead of word *x* (*at/on the double*)
- N, V — part-of-speech variables, in this case meaning all nouns and all verbs respectively (*not know one end of a(n) N from the other*)
- (...) — indicates an optional element (*an iron hand (in a velvet glove)*)

Thus, a typical entry in the LDOEI has the following form:

FINGERNAILS

hold^o/hang^o on (to^(P)) by one's fingernails/fingertips/teeth *not fml*
 to make a determined effort to keep one's position, e.g. in one's job, an activity or situation, etc.: *this country has not been pushed out of the business of building aircraft yet. It's still holding on by its fingernails* [V: often Progress]

Longman Phrasal Verbs Dictionary

The “Longman Phrasal Verbs Dictionary” (Fox 2000) is focused on one narrow type of multiword lexemes which is considered to be problematic by most foreign English learners. The difficulties are partly caused by phrasal verbs' idiomatic, non-compositional meaning. Also, they may be hard to identify (they are composed of a verb and one or two particles) and distinguish from prepositional verbs (*I looked after the kids vs the mother looked after her child going to school*).

The dictionary is organized alphabetically according to the head verbs. If several phrasal verbs use the same main verb, they are arranged according to the first particles. The most common and useful phrasal verbs are marked with a special symbol.

Apart from the definitions, the main element of the individual entries is grammatical information. Most importantly, it lists grammatical patterns for every phrasal verb, which show whether its object should be put before or after the particle (or that both possibilities are correct). Verbs that do not take any objects or take two of them are marked in a special way. Each verb is accompanied by usage examples. The dictionary also provides synonyms and opposites, information about the ability to be passivized, and related nouns or adjectives.

An entry for the phrasal verb *latch on* which takes one object after the particle or no object at all looks as follows (it is a shortened version):

latch on/onto

latch on latch on|sth

BrE informal to understand what someone means or to realize something is happening: *When they explained what kind of songs they wanted, Frank latched on really quickly.*

SIMILAR TO: catch on, cotton on

Chapter 4

Computational approach to multiword lexemes

The ability to recognize multiword lexemes is crucial for many kinds of computer applications that process natural language. This concerns, among others, computer implementations of grammars. If they are to provide correct grammaticality judgments, they need to be able to identify multiword lexemes which are often anomalous from the point of view of the given language. As far as lexicography goes, the ability to tell whether a given word is a part of a multiword lexeme or not is important for electronic dictionaries interfaced with e.g. web browsers to facilitate reading and understanding texts in foreign languages. There is not much use of a dictionary that translates *advantage* literally in the phrase *take advantage of*.

In order to recognize multiword lexemes, computational language processing tools need to have access to a lexicon of such units, which encodes them by means of some formalism, taking into account their syntactic and lexical flexibility. One way to do it is to use regular expressions which are the focus of the current chapter.

4.1. Regular expressions

4.1.1. Basic concepts

Regular expressions are formal descriptions of sets of strings. They are used by many text editors and utilities to search and manipulate bodies of text based on recurring patterns. They constitute the basic methods for string manipulation in most modern programming languages, such as Perl and Python. Their usage is common among administrators and advanced users of Unix-derived operating systems — mastering them makes most tasks related to text processing easier and less time consuming.

The concept has originated from the theories of automata and formal languages, which are fundamental for computer science. It started when Stephen Kleene described Warren McCulloch's simple automata that modelled neurons with the help of his notation called regular sets. Later on, Ken Thompson built the notation into the Unix editor `ed`. Ever since, regular expressions have been extensively used in a variety of Unix utilities.

Regular expressions are used to describe sets of strings in a concise way, without listing all the members of the given set. For example, a set of three strings: *Handel*,

Händel, and *Haendel*¹, can be described by the pattern `H(ä|ae?)ndel`. It is usually possible to describe any given set with multiple different patterns of varying complexity.

There are numerous different implementations of regular expressions, each with its own set of operators. The following set is common for most of them:

- Concatenation — this operation is marked simply by putting symbols one after another.
- Alternation — alternatives are separated with the symbol of vertical bar. For example, the pattern `gray|grey` matches *gray* or *grey*.
- Grouping — the scope and precedence of operators are defined by parentheses. For example, the strings mentioned in the previous item could also be described with `gr(a|e)y`.
- Quantification — a quantifier put after a single character or a group of them defines how many times it is allowed to occur. The most popular quantifiers are:
 - ? The question mark indicates that the previous symbol occurs 0 or 1 times. For example, the pattern `colou?r` matches both *color* and *colour*.
 - * The asterisk (aka *Kleene star*) indicates that the previous symbol can occur any number of times, including 0. For example, `go*gle` describes *ggle*, *gogle*, *google*, etc.
 - + The plus sign (aka *Kleene plus*) indicates that the previous symbol must occur at least once, but does not define any upper border. For example, `go+gle` matches *gogle*, *google*, etc., but not *ggle*.

The + and ? operators, although very useful, are redundant. They can be both expressed using only * and |.

All these simple operators can be combined, which allows one to construct complex expressions of arbitrary length, similarly to constructing complex arithmetical expressions from numbers and basic operations. For example, the pattern

```
((great )*grand)?(father|mother)
```

matches any ancestor: father, mother, grand father, grand mother, great grand father, great grand mother, etc.

4.1.2. Regular expressions and natural languages

Regular expressions are able to describe *regular languages* that are accepted by finite state automata (explained in more detail in the following subsection). They correspond to type 3 grammars in Chomsky's hierarchy. It is well-known that the syntax of natural languages cannot be described by regular, and even more complicated context-free grammars. However, there are many subsets of natural languages that can be accounted for by simple means. As it will be shown later in this chapter, it is possible to use regular expressions to describe multiword lexemes.

¹The examples of regular expressions in this section come from Wikipedia (<http://en.wikipedia.org>).

In general, “if the language² to be described is in fact regular, there may be a significant advantage in describing it by means of a regular grammar instead of using a more powerful grammar formalism” (Karttunen et al. 1996, 13). This advantage can be described in terms of 2 properties of regular expressions and regular grammars.

Firstly, parsers for such (sub)languages can be constructed directly from the regular expressions that describe them. Secondly, regular grammars can be subjected to finite-state calculus operations in order to obtain new ones without the need to rewrite them from scratch. Karttunen explains this by constructing a grammar that defines the language of dates. The grammar accepts both valid and invalid date expressions. In order to make it accept only valid ones, he creates another grammar that describes only invalid date expressions. By “subtracting” the second grammar from the first one he is able to obtain a new one that accepts only valid date expressions. Such operations would be much more difficult to perform with more powerful and complex grammars³.

4.1.3. Finite state machines

Regular languages can be visualized as special kinds of Turing machines. Turing machines are abstract devices that manipulate symbols, which were first described by Alan Turing in 1936. Such a machine consists of:

1. A tape, divided into cells, with each cell containing exactly one symbol from a finite alphabet. The tape is assumed to be arbitrarily extensible in both directions.
2. A reading/writing head that can move to the left or to the right of the tape one cell at a time.
3. A state register that contains information about the current state of the machine (the states are discrete and constitute a finite set).
4. A transition table that instructs the machine what symbol it should write, which direction it should move the head, and what its next state will be on the basis of the symbol it has just read from the tape and its current state.

Regular languages can be encoded as finite-state automata. Such automata are restricted versions of Turing machines, which can only move their head into one direction, with no going back possible (Harel 2001). Finite-state automata do not write anything — they can only accept or reject input, i.e. evaluate a given expression from the point of view of the grammar they represent.

4.2. IDioms As REgular eXpressions (IDAREX)

From the point of view of computer science, recognizing multiword lexemes is just a specific instance of a more general set of tasks that involve using regular expressions, most of which have nothing to do with processing natural languages. The IDAREX (IDioms As REgular eXpressions) formalism discussed below is an extension of regular expressions created at Xerox laboratories by Lauri Karttunen, Pasi Tapanainen, and

²Such as the language of multiword lexemes.

³Obviously, even in the case of regular grammars a working implementation of the finite-state calculus is necessary.

Giuseppe Valetto specifically for linguistic purposes (Breidt, Segond, and Valetto 1996). It has to be noted that there were other such extensions created, e.g. the query language of *Poliqarp*, a language corpus processing tool (see Chapter 5), but they are beyond the scope of this thesis.

IDAREX provides a formal way of encoding multiword lexemes with the use of regular expressions. The machinery behind the formalism that makes it possible to process such encoded units is based on Xerox’s finite-state compilers and two-level morphology whose foundations have been created by Kimmo Koskenniemi back in 1980s.

Since the time of its development, IDAREX has been employed in many diverse linguistic projects. It is a basis of context-sensitive dictionary lookup tools, such as LOCOLEX and COMPASS. It has been used in the STEEL project which aimed at creating computational tools for Eastern European languages. Moreover, it is a component of XeLDA⁴, a text processing toolkit which has been developed by Xerox Corporation and then acquired by Temis with the entire Xerox’s linguistic division.

The formalism takes the concept of regular expressions to a higher level. Instead of describing strings built of individual characters, IDAREX uses regular expressions to describe sets of phrases constructed from words — more specifically, multiword lexemes. It is based on finite-state calculus, but the computational tools are hidden under a user-friendly notation that could be described as something between ordinary regular expressions and the notation used in the “Longman Dictionary of English Idioms” (see Section 3.3.2).

Among the computational tools the most important one is a finite-state compiler which can compile IDAREX expressions into finite-state automata that computers are able to process efficiently. The formalism is independent of the underlying compiler — there were several such compilers developed by Xerox over the years. The first one was IFSM (Kattunen and Yampol), then came FSC (Tapanainen), and finally XFST (Karttunen). IDAREX was initially implemented on the basis of FSC, and then ported to XFST. However, the porting did not change the formalism itself (according to the researchers involved, only the performance of processing was optimized⁵).

IDAREX differs from Longman’s notation in several respects. The most obvious thing is that it has a working computer implementation, which to some extent proves that the formalism is coherent — a computer implementation makes it possible to test a formalism on a large amount of linguistic data, without the need for human testers who are often biased. Apart from that, IDAREX is more powerful, most notably because it enables users to define macros.

4.2.1. Xerox’s regular expressions

The regular expression formalism developed by Xerox Corporation, which IDAREX is based on, was described in Karttunen et al. 1996. It is slightly more complicated than the basics described in Section 4.1.1 because in addition to describing regular languages (i.e. sets of strings) the formalism also makes it possible to encode regular relations (i.e. mappings between two regular languages, or sets of string pairs).

Xerox’s regular expressions can be composed of two kinds of symbols. Unary symbols are used to denote strings, e.g. a regular expression of the form *a* denotes a

⁴<http://www.kazara.com/website/info/Xelda.pdf> Accessed on April 26, 2006.

⁵Information based on electronic correspondence with Kenneth Beesley and Herve Poirier (March 10, 2006).

language that contains only one string: a . Symbol pairs denote regular relations, e.g. $a:b$ is a mapping between two languages described by the regular expressions a and b .

The languages involved in a regular relation are called the upper and the lower language. Correspondingly, in the pair $a:b$ the former is the upper symbol and the latter is the lower symbol. In Xerox's formalism identity relations that map a language into itself are ignored, therefore the relation $a:a$ is abbreviated to a . The regular relations work in both directions — the upper language can be mapped (“translated”) into the lower language, and vice versa. However, by default the upper language is assumed to be the input language.

Regular relations go beyond simple finite-state automata because they require the ability to write. They can be represented by finite-state transducers. They are a special kind of finite-state machines that have two tapes instead of one. The two tapes of a transducer are typically viewed as an input tape and an output tape (the *upper* and the *lower* language respectively). Transducers are said to transduce (i.e. translate) the contents of their input tapes to their output tapes by accepting a string from the former and generating (writing) another string on the latter. This capability has obvious uses e.g. in morphology and phonology in which surface forms are derived from abstract deep representations according to a set of rules.

Xerox's formalism defines three special symbols. One of them is ϵ (epsilon) that denotes empty strings. The second one is $?$ (note that in this case it is a *symbol*, not a *quantifier*), and it stands for any symbol. The last special symbol ($\#$) is used to mark string boundaries. The special meaning of symbols can be turned off by preceding them with an escape character ($\%$) or enclosing them in double quotes.

All the basic operators listed in Section 4.1 are also valid in Xerox's formalism. In addition, it contains several ones that define more complex operations, especially those related to regular relations, but their description is beyond the scope of this thesis — for more information see Karttunen 1995 and Karttunen et al. 1996.

4.2.2. Two-level morphology

IDAREX borrows several ideas from two-level morphology which is claimed to be “the first general model in the history of computational linguistics for the analysis and generation of morphologically complex languages” (Karttunen and Beesley 2001, 1). It is a formalism for encoding morphological alterations with regular relations and expressions that can be compiled into finite-state transducers. The alterations occur between the *lexical level* (which is abstract) and the *surface level* (which represents concrete realizations of words) — hence *two-level* morphology.

The origins of the idea are related to phonological rewrite rules of the kind used by Chomsky and Halle in “The Sound Pattern of English”. Such rules involved many intermediate stages, which resulted in the formalism being asymmetric:

Traditional phonological rewrite rules describe the correspondence between lexical forms and surface forms as a one-directional, sequential mapping from lexical forms to surface forms. Even if it was possible to model the *generation* of surface forms efficiently by means of finite-state transducers, it was not evident that it would lead to an efficient analysis procedure going in the reverse direction . . . (Karttunen and Beesley 2001, 3)

Rewrite rules are unambiguous in generation, but ambiguous in analysis. A set of obligatory rules applied in a fixed order can generate only one surface form. However,

each surface form can usually be generated in more than one way. The number of possible analyses grows with the number of the rules involved.

Chomsky and Halle's rules were widely believed to be more powerful than regular relations. However, in the 1970s and the early 1980s Johnson, Kaplan, and Kay formally proved that phonological rewrite rules are no more powerful than regular relations which it is feasible to model computationally in the form of finite-state transducers.

Kimmo Koskenniemi learned about the discoveries of Johnson, Kaplan and Kay while he was visiting Xerox laboratories in the early 1980s. Back in Finland he created a new way to encode morphological alterations using finite-state methods. He broke with cascaded rewrite rules that lead to intermediate stages and computational problems. His idea was to use just two levels — one abstract and one concrete. His rules were statements that constrained lexical–surface correspondences and the environments in which they were allowed, required or prohibited. Moreover, they were applied in parallel, which eliminated the problem of intermediate stages.

The solution to the overanalysis problem has been to compile the lexicon together with the morphological rules into a single transducer. “The resulting ... transducer includes all the lexical forms of the source lexicon and all of their proper surface realizations as determined by the rules” (Karttunen and Beesley 2001, 11). Thanks to this all the ambiguities produced by the rules can be eliminated during processing.

In general, Koskenniemi's formalism is based on three ideas:

1. Rules are symbol-to-symbol constraints that are applied in parallel, not sequentially.
2. The constraints can refer to the lexical context, surface context, or both at the same time.
3. Lexical lookup and morphological analysis are carried out at the same time. The lexicon serves the purpose of a continuous lexical filter which weeds ill-formed analyses at runtime.

Apart from the general idea about representing words on two levels, IDAREX also uses the notation introduced by Koskenniemi. In the notation a colon on the right hand side of a symbol (`kick:`) refers to the lexical level, whereas a colon on the left (`:bucket`) stands for the surface level.

4.2.3. IDAREX operators and macros

All the information in this subsection has been compiled on the basis of Segond and Tapanainen 1995, Segond and Breidt 1995, and Breidt, Segond, and Valetto 1996, the first article being the most technical one, and the other two more focused on linguistic issues.

Bearing in mind the flexibility and variability of multiword lexemes, it is obvious that it is impossible to identify them automatically by simple string matching techniques. Therefore, for these purposes IDAREX uses local grammar rules encoded in the form of regular expressions:

Local grammar rules describe restrictions of MWLs [multiword lexemes] compared to general rules by implicitly stating allowed variations of the MWL compared to the default case of a completely fixed MWL. In the

default case, all restrictions apply, i.e. no variation at all is allowed, and the MWL is represented by the surface form of all lexical components in a fixed order. Violations to standard grammatical rules, e.g. missing constituents or agreement violations, need not be stated explicitly, though if necessary they can be expressed to distinguish the idiomatic from a literal use of the lexical pattern. (Segond and Breidt 1995, 3)

In the IDAREX formalism, words are represented on two levels: lexical and surface (cf. Section 4.2.2). It greatly simplifies many aspects of encoding multiword lexemes. For example, if the user wants to say that a given word is fixed and does not change, he can simply use the surface form without listing all the associated grammatical features. On the whole, there are four ways for describing individual words, all of them are marked with the way colon is used:

1. :surface-form — e.g. :house
2. :surface-form morphological variable: — e.g. :record Verb:
3. base form morphological-variable: — e.g. graduate Verb:
4. word-class-variable — e.g. ADV

The first two classes of expressions describe words that allow no variation, i.e. they cannot be inflected, modified or exchanged. The only difference between them is that the second class directly specifies the word's part of speech, which is required for ambiguous cases (e.g. *to record* and *a record*). The third class describes words that can assume various forms on the surface level. In the example above the verb *graduate* can appear in any number, tense, etc., which is indicated by the variable *Verb:* (more specific constraints are possible, e.g. restricting the verb only to its plural forms). The last class is a variable that stands for all adverbs and adverbials. Such variables are useful for encoding idioms that can be modified by a numerous group of words that would be unpractical to list individually.

A simple example that illustrates the use of both morphological levels and the morphological variables is *to take the bull by the horns*. Encoded with IDAREX, it looks like this (the * symbol has the same meaning as in basic regular expressions):

(1) ADV* take Verb: :the :bull :by :the :horns;

In the example the verb *take* can assume any form allowed by English grammar and the whole phrase can be preceded by any number of adverbials. Thus, the expression defines such instances of the phrase as *(John) took the bull by the horns*, *(John) reluctantly takes the bull by the horns* or *(They) repeatedly, reluctantly took the bull by the horns*.

The set of operators used to describe operations on words and phrases is the following⁶ (some of them perform the same function as in the case of basic regular expressions):

- *nothing* — words succeed each other

⁶The operators presented here are used for encoding multiword lexemes. However, the compilers used for processing IDAREX expressions use a much more sophisticated finite-state calculus with many more complex operators which are described in more detail in Segond and Tapanainen 1995.

- *, +, | — cf. Section 4.1.1 above
- parentheses () — mark an optional part of the idiom
- brackets [] — group an expression
- semicolon ; — finishes an expression

Using the operators and the two-level representation it is possible to create much more complicated descriptions than (1). The following expressions illustrate just a few possibilities:

- (2) [:at | :on] :the :double;
- (3) :an :iron [:hand | :fist] (:in :a :velvet :glove);
- (4) get Verb: N :off NPOSS :chest;
- (5) :not know Verb: :one :end :of [:a|:an] N :from :the :other;

Expression (2) is relatively simple with just a single alternative. It matches exactly two phrases: *at the double* and *on the double*. Expression (3) is slightly more complicated: it contains an alternative and an optional part, which on the whole gives four possible phrases. There are two word-class variables in expression (4), one matches all nouns and the other all possessive nouns; as a result the expression matches an infinite number of phrases (although some of them might make no sense semantically). Expression (5) demonstrates the use of alternative for the determiner, in order to use a correct one for the following noun (appropriate constraints for using *a* and *an* must be defined elsewhere).

IDAREX has one more very powerful feature which is the possibility to create macros. Macros allow linguists to capture syntactic generalizations in a concise and general way, without the need to define them separately for all individual cases.

A multiword lexeme that can illustrate the idea is *to pass the hat round*. According to the “Longman Dictionary of English Idioms”, the phrase can also appear in a form in which the preposition *round* is placed directly after the verb: *to pass round the hat*. One could encode both possibilities with IDAREX as follows:

- (6) [pass Verb: :the :hat :round | pass Verb: :round :the :hat];

However, there are other multiword lexemes that follow the same pattern and it would be necessary to define such an alternative for each one of them. This would be greatly inefficient, therefore macros were devised to handle such cases.

In order to account for the multiword lexeme above and all the similar cases, one could define a macro called “Particle movement”:

- (7) Particle_movement:
 [\$1 Verb: (Adj*) N \$2 Prep: | \$1 Verb: \$2 Prep: (Adj*) N]

The variables marked with \$ serve the purpose of “slots” that can be filled with any word that belongs to the defined class. The most important thing here is the indices which force the usage of the same words on both sides of the alternative — if \$1 is filled with, e.g. *pass* on the left side of the expression, it has to be *pass* also on the

right side. The same goes for variable \$2. I have added an optional (Adj*) element to the macro, in order to make it more general and be able to match a larger number of phrases (of course, this does not guarantee the broadest coverage possible; there should also be ADV variables added in order to account for adverbials).

Another powerful feature of IDAREX is related to the finite-state machinery which it is based on. As it has been mentioned earlier, the technology allows to add, intersect, and subtract the finite state networks compiled from regular expressions. For example, if the user wanted his finite-state network to be more restrictive, he could write some rules for the phrases that should *not* be accepted, compile them into a finite-state network, and then subtract the latter from the former network — see (Karttunen et al. 1996) for more details.

The formalism also has some drawbacks. One is that the local grammar rules are formulated to be as general as possible, which makes overgeneration possible. The rules could be made more specific and restrictive, but the formalism’s creators have assumed that there can be “no ill-formed input” (Segond and Breidt 1995, 6). For practical applications it does not really matter if the rules allow for more variation that can actually appear in a text, as long as the idiomatic and the literal uses of a phrase can be distinguished.

The other issue is that the formalism is unable to account for productive variations of multiword lexemes, created ad hoc by language users. These, however, are unforeseeable by their very nature.

Yet another problem is related to variations that are heavily based on varying word order, as it is the case with German verbal multiword lexemes. Accounting for many possible verb complements that can appear in various cases, with topicalization and other transformations allowed, requires writing very complex regular expressions which are not only hard to read, but also to process effectively (Segond and Breidt 1995, 10). In such cases regular expressions reach the limit of their expressiveness.

4.3. Practical applications of IDAREX

Potential applications of regular expressions, and more specifically IDAREX, are very numerous and diverse. According to some authors, they range from intelligent dictionary lookup, over concordancing and indexing, to machine translation (Breidt, Segond, and Valetto 1996). It is hard to establish in how many commercial products IDAREX has been used as a part of the XeLDA toolkit.

The following sections focus on applications that have to do with lexicography, and more specifically electronic dictionaries. Thanks to IDAREX lexicographers are able to go beyond “ordinary” dictionaries that the user can query for definitions or translations of single words, and create *comprehension assistants* (see below) which feature *context-sensitive dictionary lookup*. Several such tools are described in the following sections.

4.3.1. LOCOLEX

LOCOLEX is an electronic dictionary engine which its authors dubbed “an intelligent reading aid” (Bauer, Segond, and Zaenen 1995, 1). According to them, one of the greatest obstacles on the way to understanding a text in a foreign language, regardless of the reader’s level of skill, is encountering an unknown word or phrase which requires

dictionary lookup. This is supposed to result in frustration, because searching manually in a paper dictionary can last up to several minutes and requires the primary task to be put aside. LOCOLEX is said to provide “intelligent dictionary lookup through the interaction between a complete online dictionary together with online text.”

The authors of the system identified three issues that bother users of traditional paper dictionaries:

1. In most cases the reader does not need access to full dictionary entries in order to understand a text, a list of direct equivalents is often sufficient.
2. It is often the case that the reader encounters an inflected verb whose base form he does not know and thus cannot easily look it up in a dictionary (e.g. *slew* can be either a noun or the past form of the verb *to slay*; looking the word up in its original form would only give the definition of the noun).
3. The last issue is something that has been already mentioned — if the unknown word is a part of a multiword lexeme the reader does not know, looking up the word alone does not result in better comprehension.

The reason for calling LOCOLEX “intelligent” seems to be the fact that it addresses all those issues in order to make dictionary lookup easier and less distracting.

As far as the first issue goes, initially LOCOLEX displays only a brief list of translations for the given word. However, unlike simple translation dictionaries, it does contain all the additional information. Therefore, if the reader finds the translation list inadequate, he can browse the individual translations and access all the information like pronunciation, definition, usage examples, etc.

In the case of queries for inflected words, LOCOLEX performs morphological analysis and returns the word’s base form. However, this can still lead to ambiguous results (e.g. *records* can be analyzed as the third person form of the verb *to record* or the plural form of the noun *record*). To remedy it, the system examines the context in which the word appears. If the word is positioned between an adjective and a preposition, the noun interpretation will be given precedence.

Context is also used to determine whether the queried word is a part of a multiword lexeme — this is where IDAREX comes into play. LOCOLEX analyzes the context and compares it against the regular expressions that encode multiword lexemes. As it has been described above, the regular expressions in IDAREX cover inflection and other kinds of variation that can appear in such units, therefore the system should have no problems with identifying idioms and compounds in real texts. The grammar rules (i.e. the regular expressions) add a “dynamic” element to the traditionally static dictionaries. The dictionary — or rather the “intelligent reading aid” — reacts to context and actively assists the reader in the lookup process, instead of listing the translations or the definitions in a way that is too often inaccurate and not very helpful. Unfortunately, the authors do not provide any information regarding the system’s accuracy in identifying multiword lexemes.

A more detailed description of all the stages involved in a single dictionary lookup via LOCOLEX is presented in Breidt and Feldweg 1997. The whole process is quite complicated and consists of the following:

1. Tokenization — upon a user’s request for a word, LOCOLEX isolates the sentence containing the word and breaks it down into individual words (*tokens*).

2. Morphological analysis — each word undergoes morphological analysis with a two-level analyzer. As a result, each word is stripped to its base form and assigned a set of morphosyntactic labels. Ambiguous words are assigned with multiple analyses.
3. Normalization — words that were not recognized by the morphological analyzer are fed into a specialized transducer which examines them from the point of view of common orthographic variations. After this step the words are once again sent to the morphological analyzer.
4. Guesser — if the analyzer still cannot recognize the words, they are assigned morphosyntactic information by the guesser which uses heuristics to determine which category do the words belong to.
5. Part-Of-Speech disambiguation — at this stage all the morphosyntactic ambiguities are resolved by means of statistical methods.
6. Dictionary access — the base form of the requested word is looked up in a pre-compiled dictionary index in order to locate the appropriate entry. To speed up the whole process, the dictionaries are stored in an internal, binary format (similarly, all the IDAREX expressions are stored in compiled form together with the dictionaries).
7. Part-Of-Speech mapping — POS information is used to determine which part of the entry is relevant for the user in a particular situation. Before marking an appropriate subentry, differences between the POS tags of the analyzer/disambiguator and the dictionary are resolved by a POS mapping transducer which maps the category chosen by the disambiguator to the POS labels of the dictionary.
8. Context evaluation — at this stage LOCOLEX analyzes the word's context and looks for strings that match IDAREX expressions in order to identify multiword lexemes.

The system's creators propose to enhance LOCOLEX to make better use of the usage labels present in dictionaries. The idea is that the reader, who probably is at least generally aware what field the text is concerned with, should choose several topics, and then the reading aid would rank possible translations according to the usage labels and filter out the meanings that are inappropriate in the given context. The idea is interesting, although it is hard to predict whether it would prove to be useful should it ever be implemented.

4.3.2. COMPASS

COMPASS (COMPrehension ASSistant) was a European project conducted between 1994 and 1996 (Breidt and Feldweg 1997). Among its most notable members were the French research center of Xerox (RXRC) and University of Tuebingen. The project's main premise was the idea that people were encountering more and more texts in electronic form, which were usually not important enough to deserve proper translation. In most cases people only wanted to acquire some general understanding of the text. Therefore, researchers behind the project decided to create an electronic "comprehension assistant" — an aid for reading texts in electronic form which would be something

more than a simple dictionary, but also less than a full-featured machine translation system.

As in the case of LOCOLEX, the idea was to create a tool that would release the readers from the burden of manual dictionary lookup and enable a seamless reading experience. However, the authors also claim that it can be used to support “language learning ‘on the fly’, building on the users’ existing knowledge of the foreign language.”

All these functions are based on computational methods of natural language processing and a workflow similar to LOCOLEX: morphological analysis and determining the base form, determining the part of speech, checking for multiword lexemes (this is done with the help of IDAREX), and displaying only those parts of the entry that are relevant for the word in the given context. A diagram of the system’s organization reveals that actually all these functions are performed by LOCOLEX which COMPASS uses as a backend. The things that were actually created in the project included a graphical user’s interface, “language models” (i.e. lexicons, information for part of speech disambiguation, IDAREX expressions, etc. for individual languages), and bilingual electronic dictionaries⁷.

4.3.3. STEEL

The STEEL (Developing Specialized Translation/Foreign Language Understanding Tools for Eastern European Languages) project was conducted in the late 1990s and involved the technologies developed by Xerox, most notably IDAREX and LOCOLEX. One of the aspects of the project was the creation of electronic dictionaries in the XeLDA format. The Warsaw University team prepared one on the basis of a traditional dictionary by Piotrowski and Saloni (1997)⁸. The most interesting about the project was the way in which the dictionary had been converted into electronic form and enhanced with IDAREX expressions, which is described in Piotrowski 1999 and Głowińska and Woliński 2000.

The internal format for all XeLDA dictionaries was SGML (Standard Generalized Markup Language), which has been replaced in modern applications by XML (eXtensible Markup Language). Saloni and Piotrowski’s dictionary was initially encoded in \TeX which is a language intended rather for encoding typographical information, and not structure. However, if used in a reasonable way, \TeX can quite successfully be used for structural markup. A sample entry of the dictionary’s original \TeX format is presented in Figure 4.1. The printed result is shown in Figure 4.2. The markup is based on mostly unambiguous structural tags, even though simple typographical tags could be used (e.g. italics for examples, bold for the headword, etc.) However, in the latter case, it would be extremely difficult to convert the dictionary into any other structural markup format except for manually. Because the \TeX markup was done properly, it was possible to convert the dictionary into SGML automatically. Firstly, a grammar for the dictionary’s markup was prepared and its mapping into the XeLDA DTD (Document Type Definition). Then the actual conversion took place. Automatic conversion was successful for ca. 95% of the data. The problematic cases were the most commonly used words, such as *have*, whose entries are very complex. A resulting SGML entry, containing an IDAREX expression, is shown in Figure 4.3.

⁷The most interesting thing about the COMPASS project seems to be the way in which printed dictionaries were converted into SGML from magnetic type-setting tapes. However, this goes beyond the scope of this thesis. See Breidt and Feldweg 1997.

⁸Only the English-Polish section was used.

```

{{\haslo pace}
[{\wym peis}]
{\nr 1.} tempo
{\nr 2.} krok
{\nr 3.} chodzi"c ({\lacz sth} po czym"s)
$\diamond$ {\nr 4.} {\idiom keep pace with sth} dotrzymywa"c czemu"s kroku
{\nr 5.} {\idiom at a (good) pace} (dobrym) tempem
{\nr 6.} {\idiom do sth at one's own pace} robi"c co"s w"lasnym tempem
{\nr 7.} {\idiom take (two) paces} przej"s"c/zrobi"c (dwa) kroki
{\podhaslo pace maker}
[{\wym {\'}peismeik{\e}}]
stymulator serca
}

```

Figure 4.1: A sample entry from Saloni and Piotrowski's dictionary encoded in T_EX.

```

pace [peis] 1. tempo 2. krok 3. cho-
dzić (sih po czymś) ◇ 4. keep
pace with sth dotrzymywać cze-
muś kroku 5. at a (good) pace
(dobrym) tempem 6. do sth at
one's own pace robić coś wła-
snym tempem 7. take (two)
paces przejść/zrobić (dwa) kroki
pace maker ['peismeikə] stymu-
lator serca

```

Figure 4.2: An entry in the printed version of Saloni and Piotrowski's dictionary.

The most important properties of the XeLDA dictionary format are the following:

1. the entries are composed of three levels: lexemes (**syntactic**), meanings (**semantic**) and equivalents (**subsense**);
2. the **senseinfo** sections explicitly state whether the information they contain is related to the English entry, or the Polish equivalent;
3. every single translation is marked with separate **trans** tags;
4. there is a distinction between stylistic (**label**) and language-related (**lang**) qualifiers.

Because XeLDA is equipped with a tokenizer and a morphological analyzer, there is no need to include entries for inflected words whose sole purpose is to point to other parts of the dictionary.

For several reasons the original dictionary lacked POS labels (Piotrowski 1999). However, since the labels are very important for LOCOLEX, because it uses them to select the parts of entries to be displayed (cf. Section 4.3.1), they had to be added to the dictionary manually.

For the purposes of XeLDA, all multiword lexemes in the dictionary had to be encoded in IDAREX. Most of this was done automatically — English multiword lexemes were extracted from the dictionary and processed with XeLDA tools. However, the task involved two difficulties. Firstly, some of the idioms in the original dictionary were classified as such because of the author's design decision, and did not count as

```

<entry>
  <headword>
    <spl>pace</spl>
  </headword>
  <hwninfo>
    ...
  </hwninfo>
  <syntactic>
    <senseinfo>
      <pos>N</pos>
    </senseinfo>
    <semantic>
      <subsense>
        <trans>tempo</trans>
      </subsense>
    </semantic>
    ...
  </syntactic>
  <syntactic>
    <senseinfo>
      <pos>V</pos>
    </senseinfo>
    <semantic>
      <subsense>
        <trans>chodzić</trans>
        <preposition>
          <prepsource>sth</prepsource>
          <preptarget>po czymś</preptarget>
        </preposition>
      </subsense>
    </semantic>
    ...
    <semantic>
      <subsense>
        <idiom>do sth at one's own pace</idiom>
        <idarex>do V: NP :at PROPOSS :own :pace</idarex>
        <trans>robić coś własnym tempem</trans>
      </subsense>
    </semantic>
    ...
  </syntactic>
  ...
</entry>

```

Figure 4.3: A sample, incomplete SGML entry from Saloni and Piotrowski's dictionary containing an IDAREX expression.

idioms for the purposes of IDAREX. Secondly, the canonical forms of idioms in the dictionary were often different from the ones recognized by XeLDA, which made automatic recognition difficult or impossible. Such cases were processed manually by human lexicographers with the help of GNU Emacs⁹.

The conversion process unveiled the areas in which dictionaries meant for human users fail to satisfy computational requirements. In order to be user friendly and save space, the dictionary employed various graphic symbols, such as asterisks and slashes, to introduce variants, mark their boundaries, etc. These symbols had to be rewritten into appropriate, unambiguous tags. Also, the dictionary contained many redundant elements which the XeLDA system was able to generate on its own. These included information about inflection and dialect variants. Since the DTD of XeLDA did not allow for such elements to be included, they had to be removed or placed in other fields within the structure.

4.4. Similar applications

4.4.1. MoBiMouse

MoBiMouse is an electronic dictionary lookup application developed by MorphoLogic¹⁰, a Hungarian company based in language technology industry, which took part in some of the projects based on Xerox's technology. The application's significance lies in the fact that it is the only available (also commercially) electronic dictionary interface that offers context-sensitive dictionary lookup and multiword lexeme recognition. MoBiMouse is based on several assumptions that are close to those of LOCOLEX and COMPASS (Prószéky and Földes 2005). Its creators similarly believe that translation and foreign text comprehension are two different things. According to them, a seamless reading experience requires that the dictionary lookup process should be as unobtrusive as possible. The electronic dictionary should provide the reader with the absolute minimum information required for understanding, and do it as quickly as possible, preferably with little or no interaction from the reader.

MoBiMouse, which its authors dubbed a "context-sensitive instant comprehension tool" (Prószéky and Kis 2002), tries to fulfill all these requirements. First of all, its interface tries to be as user-friendly as possible. In order to look up a word, the user only has to position the mouse pointer over it. There is no clicking and no menus involved. It is possible to customize the time interval between pointing to a word and the actual lookup taking place. One of the things that makes MoBiMouse different from COMPASS is that it does not behave like a separate application. It is integrated with the whole desktop environment, thanks to which it is possible to lookup words inside any application "natively" — it is not necessary to copy text into the window of the comprehension assistant. The program uses the operating system¹¹ API (application program interface) to reach the text or an OCR (optical character recognition) procedure when the former is impossible.

After the program reaches the word to be looked up, linguistic analysis is performed. It consists of stemming, spelling correction, and shallow parsing of the context during which multiword lexemes are recognized. Unfortunately, the authors do not reveal

⁹<http://www.gnu.org/software/emacs>

¹⁰<http://www.morphologic.hu>

¹¹MoBiMouse is available only for MS Windows.



Figure 4.4: MoBiMouse displays its stemming capabilities.

the details about the process of multiword lexeme recognition in any of their articles, and when queried about it personally. The only certain thing is that they do not use IDAREX, because at one point they say that “the treatment of multiword units in the IDAREX formalism . . . is more sophisticated than in MoBiMouse” (Prószéky and Kis 2002). Most probably they use some kind of regular expressions.

Dictionary lookup takes place after the linguistic analysis is finished. Another difference between MoBiMouse and COMPASS is that the former can look up words in several dictionaries¹² simultaneously, including custom ones prepared by the users, and not in only one at a time. The whole procedure, from the time the user points to a word, to the time the dictionary lookup results are displayed to him, is said to take only about 0.02 s on average (although it is not indicated what kind of computer this figure is based on). Figure 4.4 shows MoBiMouse displaying the results of stemming.

On the whole MoBiMouse does not seem to go beyond LOCOLEX and COMPASS as far as language technology is concerned (at least this is what can be judged from the available materials). The application’s focus seems to be on speed of processing and user interface features. Its integration with the whole desktop environment and the ability to look up words in any open application are certainly innovative and indicate the direction which the development of electronic dictionaries should turn to.

¹²The Polish-English dictionary described in Prószéky and Földes 2005 is based on the already mentioned work by Piotrowski and Saloni.

Chapter 5

Towards a free implementation of context-sensitive dictionaries

There do not seem to be any projects related to IDAREX going on at the moment, especially in the research field. Also LOCOLEX and COMPASS seem to be inactive and not developed anymore. As it has been said earlier, IDAREX has been just a special implementation of regular expressions. Currently it seems that it died together with Xerox's finite-state tools that it was based on¹. Moreover, the only dictionary tool that provides the functionality of IDAREX, which is MoBiMouse, is only available for MS Windows operating systems.

Therefore, in the last chapter, I would like to postulate creating a free² implementation of a similar extension to regular expressions which could be used for further research on multiword lexemes and for creating free, multiplatform, context-sensitive dictionaries. The following sections describe several potential research projects concerning multiword lexemes and context-sensitive dictionaries, including a proposal to extend the DICT protocol for handling IDAREX expressions. All of them are based on ideas which are courtesy of prof. J. S. Bień.

5.1. Extending the DICT protocol

5.1.1. DICT — a dictionary server protocol

The DICT protocol³ has been first described by Rickard E. Faith and Bret Martin in October 1997 as a replacement for the “webster” protocol. At the time the RFC was written, several free dictionaries (such as the Jargon file and Wordnet) became available which it was impossible to handle conveniently with “webster”, because it could only provide access to one dictionary at a time.

From the very beginning DICT has been intended to handle multiple lexical databases simultaneously. The protocol is designed on a client-server basis and makes it possible to serve the databases in a local network or via the Internet. It enables clients to

¹It has to be emphasized that the tools are dead because of the company's policy, rather than because they were bad software. Xerox does not maintain them anymore and has not opened their code, which might have resulted in further development by the linguistic community.

²As in *freedom*.

³RFC 2229 <http://www.faqs.org/rfcs/rfc2229.html> An RFC (Request For Comments) is one of a series of documents that traditionally describe new technologies and methodologies related to the Internet.

request word definitions, search the word index, and request information about the server and the individual dictionaries. It is designed to handle Multipurpose Internet Mail Extensions (MIME) which make it possible to transmit binary data via ASCII-only protocols. Thanks to this, dictionaries can potentially contain any kind of data, including sound, pictures, movies, etc.

The protocol does not define the structure of the databases — it is a responsibility of programmers who implement DICT servers. This makes it possible to develop servers that can handle any format, e.g. flat text or TEI⁴. It is also possible to implement custom search strategies (which are basically algorithms for searching the databases). The most popular strategies are:

- exact — match headwords exactly (required for all servers)
- prefix — match prefixes (required for all servers)
- regexp — basic regular expressions

The communication between the client and the server, as defined in the RFC, is relatively simple. Following is a description of two commands used to retrieve information from a DICT server. All the communication with the server is presented in raw format, such as can be viewed when connecting to the server with `telnet`.

1. The `DEFINE` command requests the server to lookup up a specified word in a specified database. Its syntax is the following:

```
DEFINE database word
```

It is possible to use `!` for the name of the database to make the server look through all the available dictionaries and return the first match, or `*` which makes the server return matching definitions from all available databases. If the server finds any matching definitions, it returns an appropriate status code and sends the queried word, the name of the database, a short description of the database, and the relevant definition(s), as it is shown in the following transaction (the output has been shortened):

```
220 chiba.nippon.net dictd 1.10.4/rf on Linux 2.6.16.13-sendai
<auth.mime> <3.4927.1146906156@chiba.nippon.net>
```

```
define jargon "laser chicken"
```

```
150 1 definitions retrieved
151 "laser chicken" jargon "Jargon File (4.3.1, 29 Jun 2001)"
```

```
laser chicken n. Kung Pao Chicken, a standard Chinese dish
containing chicken, peanuts, and hot red peppers in a spicy
pepper-oil sauce. Many hackers call it 'laser chicken' for
two reasons: It can zap you just like a laser, and the sauce
has a red color reminiscent of some laser beams. The dish has
also been called 'gunpowder chicken'.
```

```
.
250 ok [d/m/c = 0/13/239; 0.000r 0.000u 0.000s]
```

⁴Text Encoding Initiative <http://www.tei-c.org>

2. The `MATCH` command searches the index of a specified dictionary and returns all the words that were matched using a particular search strategy. The syntax is:

```
MATCH database strategy word
```

Similarly to the `DEFINE` command, it is possible to use wildcards for the database name. If `.` is used for the name of the strategy, the server searches the databases with its default algorithm. After the search is made, the server returns an appropriate status code and sends a list of matched words:

```
220 chiba.nippon.net dictd 1.10.4/rf on Linux 2.6.16.13-sendai
<auth.mime> <3.4927.1146906156@chiba.nippon.net>

match foldoc regexp "^unix.*"

152 13 matches found

foldoc "unix"
foldoc "unix box"
foldoc "unix brain damage"
foldoc "unix conspiracy"
foldoc "unix international"
foldoc "unix man page"
foldoc "unix manual page"
foldoc "unix system v"
foldoc "unix to unix copy"
foldoc "unix weenie"
foldoc "unix wizard"
foldoc "unixism"
foldoc "unixware"
.
250 ok [d/m/c = 0/13/239; 0.000r 0.000u 0.000s]
```

It has to be remembered that the `MATCH` command might return several hits for each database, especially when used with regular expression strategies, such as it has been shown above.

5.1.2. Multiword lexemes and DICT

The following is a proposal to extend the DICT protocol in order to make it possible to use IDAREX or similar expressions with it. It has to be noted that it is just a protocol extension proposal, and therefore no claims are made concerning the structure of the actual dictionaries or the implementation of the search algorithms involved.

The proposal is based on three assumptions:

1. Bearing in mind the variability of multiword lexemes, it is obvious that in many cases effective lookup would require morphological analysis to be performed. The protocol should enable the process to be done either on the client or the server side.

2. The implementation of a multiword lexeme formalism should be a part of the client, and therefore it is the client that will be responsible for checking whether the expressions returned by the server match the queried word or phrase.
3. A new MIME extension should be defined, e.g. `x-idarex-list`, for files containing lists of expressions that describe multiword lexemes. That is to make the client aware that the result is a list of expressions, and not e.g. a definition.

The following describes a hypothetical transaction between a client and a DICT server with multiword lexeme lookup support on the basis of the phrase *lost his head*.

1. The client sends a `MATCH` command to the server:

```
MATCH * idarex lost
or
MATCH * idarex {lose}
```

The first query sends an inflected word and requests the server to perform morphological analysis. The curly brackets in the second query indicate that the word has been already transformed to its base form. The strategy is `idarex`, which should search the dictionary for any IDAREX-like expressions which the word in question can be a part of, and return a list of potential matches. The list should be compatible with the `x-idarex-list` MIME type. Alternatively the server could indicate that the response contains IDAREX expressions by returning an appropriate response code reserved specifically for this purpose. This seems to be simpler, but the final decision should be based on a careful analysis by the programmers carrying out the implementation.

2. Upon receiving the list of expressions, the client runs language technology tools in order to analyze the context of the specified word and determine which of the returned expressions is the correct one (if any). The reason this should be done on the client side is that it is impossible to predict how much context will be necessary for processing the query — the client has access to the source of the queried word, whereas the server does not, even if run locally. If the client finds a matching expression, it sends another `MATCH` command to the server with the expression's number in the list:

```
MATCH * idarex 'lose:3'
```

Alternatively, it could be a `DEFINE` command (without specifying the `idarex` strategy), but this should probably be resolved after designing the structure of the dictionaries in question.

3. As a response, the server should send the client an appropriate part of the relevant dictionary entry, containing a definition or a translation of the multiword lexeme.

The most important thing to do, apart from implementing a formalism similar to IDAREX, would be to transform words to their base forms, i.e. lemmatize them. Nowadays this can be done with the help of freely available tools. Simple lemmatizers for English can be created with the Natural Language Toolkit⁵ which is a set of

⁵<http://nltk.sourceforge.net>

Python libraries for building natural language processing tools. Even the basic Unix spellchecker, `ispell`, offers lemmatizing capabilities. Obviously, this would be useful not only for querying for multiword lexemes, but inflected words in general.

The question of whether the lemmatizing should be done on the client or on the server side should be left to the programmers, but it seems that making the client do it is a better solution. For one thing, servers should be kept as simple as possible for security and performance reasons. Secondly, if a server were handling dictionaries for many languages, it would probably be very hard to implement morphological analysis functions for all of them in a single server. Even if morphological analyses were performed by some external processes, the machine that the server runs on would need to have access to several such tools, one for each language involved. Thus, leaving it to the client should result in greater simplicity and reliability.

5.2. Reusing Piotrowski and Saloni's dictionary

An appropriate dictionary for usage with a DICT server modified to handle the above is the English-Polish dictionary mentioned in Section 4.3.3 (Piotrowski and Saloni 1997). Its authors have declared that it might potentially be released under the terms of the GNU General Public License. In 2006 it has been converted into the TEI format by Adam Mazur and Maciej Wojciechowski as a final assignment for the computer lab classes accompanying prof. J. S. Bień's *Słowniki elektroniczne — budowa i użytkowanie*⁶ lectures held at the Institute of Informatics at Warsaw University. The dictionary contains 2216 IDAREX expressions⁷ encoded as `<note type="idarex">` elements.

Even without lemmatizing or any finite-state technology, Saloni and Piotrowski's dictionary can serve the purpose of a useful database of multiword lexemes. Among the 2216 IDAREX expressions it contains, there are only 1062 with words on the lexical level, i.e. ones that can be inflected. The remaining 1154 expressions include 49 that have optional elements, 103 that contain an alternative, and 163 that contain POS variables (some expressions combine these elements). On the whole the dictionary has 875 IDAREX expressions which describe absolutely frozen multiword lexemes that allow no variation. These can be identified within texts with simple string matching techniques.

5.3. Verifying a multiword lexeme formalism on a corpus

If a free implementation of an IDAREX-like formalism were available, it would provide a strong motivation to verify the adequacy of describing multiword lexemes with regular expressions on a language corpus. The whole process would require three things: an appropriately sized linguistic corpus, a corpus processing tool, and an algorithm for translating the language of the multiword lexeme formalism into the query language of the corpus processing tool.

⁶*Electronic dictionaries — structure and usage.*

⁷All the subsequent figures have been obtained by running basic Unix tools (`grep`, `wc`, `cat`, `cut`, `diff`, `sort`, and `uniq`), with a tiny bit of manual tweaking.

A good choice for the corpus processing tool is *Poliqarp*⁸ which is a suite of software that has been originally developed for the purposes of a corpus of Polish created at the Institute of Computer Science of the Polish Academy of Sciences. It can be used to effectively search large language corpora using a highly sophisticated query language (Przepiórkowski et al. 2004). It is this language that IDAREX expressions would have to be translated to. Alternatively, one could consider using *Sara* or *Xaira*, which are tools developed for the purposes of the British National Corpus⁹.

As far as the corpus goes, the project would require it to be annotated with morphological information. Only with such a corpus it would be possible to run queries translated from IDAREX expressions containing words on the lexical level or POS variables. The following expression can serve as an example:

(1) `be V: (ADV) :angry :with`

Translated into a *Poliqarp* query, it would assume the following form:

(2) `[base='be'] [pos=adv] ?[orth='angry'] [orth='with']`

This query searches for all forms of the word *be*, followed by an optional, single adverbial, and the string *angry with*. All this requires the words in the corpus to carry information about their lemmata and POS categories.

Some candidates for corpora that could be used in the experiment are the British National Corpus, and the corpora distributed with the LinGo¹⁰ grammar. Some corpora snippets are also distributed with the Natural Language Toolkit.

In the worst case, an appropriate corpus could be build from scratch. One step in the process would require using Buczyński's *Kolokacje* which, apart from finding collocations, can function as an effective web crawler for collecting textual data from Internet pages. Once collected, the data could be linguistically annotated with an appropriate piece of software, such as Eric Brill's rule-based tagger¹¹. It is also possible to build a simple tagger from scratch using the predefined functions distributed with the Natural Language Toolkit.

⁸<http://poliqarp.sourceforge.net>

⁹<http://www.natcorp.ox.ac.uk>

¹⁰<http://lingo.stanford.edu>

¹¹<http://www.cs.jhu.edu/~brill/>

Chapter 6

Conclusions

Computer technology has had a great impact on the methodology and the scope of interest of linguistics. An example of the process is the treatment of multiword lexemes, which evolved from very general and inconsistent descriptions like the one in (Smith 1943) into fully formalized models, such as IDAREX.

Achieving greater accuracy of linguistic description is a utilitarian benefit of combining linguistics with computer science — a more important one is that technology enables linguists to explore new areas of research. Corpus linguistics, statistical study of language, and building ontologies were all impossible without adequate computational tools. Such tools are now widely available in the form of free operating systems and open source linguistic software which allow everyone to experiment with the latest technologies and contribute to developing them. Obviously professional research takes more than mere experimenting, but the educational potential of GNU/Linux or the Natural Language Toolkit cannot be overlooked.

Combining linguistics with computer technology seems to be very promising for lexicography. Language corpora allow dictionary authors to access huge amounts of linguistic data in an instant and search through it effectively, thanks to sophisticated query languages and linguistic annotation. Interesting new phraseologisms can be found by means of statistical methods implemented in tools such as *Kolokacije*. Marking up lexicographic data with XML and storing it in the form of computer files makes it possible to produce multiple versions of a dictionary from one source, both in printed and electronic form, which saves time and resources.

Computational lexicography has also altered, and in most cases improved, the way in which users access their dictionaries. They are no longer limited to searching words manually according to an alphabetic order of entries — instead they are provided with sophisticated electronic dictionaries which offer incremental, regular expression, and a *tergo* searches.

The Internet and protocols such as DICT might some day make it unnecessary to possess physical copies of dictionaries, as they allow users to access electronic versions of lexicographic works from anywhere in the world. The most interesting developments in this context are those related to *Wikipedia* and *Wiktionary* which blur the distinction between the lexicographer and the user. Open, wiki-based dictionaries are not likely to replace the ones prepared by professional lexicographers. However, the ideas of collaborative work and lexicographer-user interaction will undoubtedly become a part of the dictionary compilation process, which is already being realized by some publishing houses.

Bibliography

- Bauer, Daniel, Frédérique Segond, and Annie Zaenen. 1995. "LOCOLEX: The Translation Rolls off Your Tongue." *Proceedings of ACH-ALLC*. Santa Barbara, CA.
- Benson, Morton, Evelyn Benson, and Robert F. Ilson. 1986. *Lexicographic Description of English*. Amsterdam/Philadelphia: John Benjamins.
- Breidt, Elisabeth, and Helmut Feldweg. 1997. "Accessing Foreign Languages with COMPASS." *Machine Translation* 12 (1/2): 153–174.
- Breidt, Elisabeth, Frédérique Segond, and Giuseppe Valetto. 1996. "Formal Description of Multi-Word Lexemes with the Finite-State Formalism IDAREX." *Proceedings of the 16th Conference on Computational Linguistics*, Volume 2. Morristown, NJ: Association for Computational Linguistics, 1036–1040. <http://acl.ldc.upenn.edu/C/C96/C96-2182.pdf>.
- Brundage, Jennifer, Maren Kresse, Ulrike Schwall, and Angelika Storrer. 1992. "Multiword Lexemes: A Monolingual and Contrastive Typology for NLP and MT." Technical Report IWBS 232, IBM Deutschland GmbH, Institut für Wissenbasierte Systeme, Heidelberg.
- Buczyński, Aleksander. 2004. "Pozyskiwanie z Internetu tekstów do badań lingwistycznych." Master's thesis, Warsaw University. <http://www.mimuw.edu.pl/polszczyzna/kolokacje/doc/pozyskiwanie-tekstow.pdf>.
- Cruse, D. A. 1986. *Lexical Semantics*. Cambridge: Cambridge University Press.
- Cutler, Anne. 1982. "Idioms: The Colder the Older." *Linguistic Inquiry* 13:317–320.
- Fillmore, Charles J., Paul Kay, and Mary Cathrine O'Connor. 1988. "Regularity and Idiomaticity In Grammatical Constructions: The Case Of *let alone*." *Language* 64 (3): 501–538.
- Fox, Chris, ed. 2000. *Longman Phrasal Verbs Dictionary*. Harlow: Pearson Education Ltd.
- Fraser, Bruce. 1970. "Idioms within a transformational grammar." *Foundations of Language*, no. 6:22–42. Quoted in (Reagan 1987).
- Gates, Edward. 1988. "The treatment of multiword lexemes in some current dictionaries of English." Edited by Mary Snell-Hornby, *ZuriLEX '86 Proceedings*. Tübingen: Francke, 99–106.
- Grosbart, Herman. 1987. "Szkic teoretycznych założeń projektu komputerowego słownika przekładowego i prozopozycja podjęcia prac nad prototypowym komputerowym słownikiem rosyjsko-polskim." In *Studia z polskiej leksykografii współczesnej*, edited by Zygmunt Saloni, Volume II, 287–307. Dział Wydawnictw Filii UW w Białymstoku. <http://www.mimuw.edu.pl/polszczyzna/Grosbart-S87/Grosbart-S87.pdf>.

- Guenther, Frantz, and Xavier Blanco. 2004. “Multi-lexemic Expressions: an Overview.” In *Syntax, Lexis, and Lexicon-Grammar*, edited by Christian Lèclere; Éric Laporte; Mireille Piot; Max Silberztein, Volume 24 of *Linguisticae Investigationes Supplementa*, 239–252. John Benjamins. <http://seneca.uab.es/filfrirom/BLANCO/PUBLIC/multilex.pdf>.
- Głowińska, Katarzyna, and Marcin Woliński. 2000. “Angielsko-polski słownik elektroniczny XeLDA.” *Acta Universitatis Nicolai Copernici. Studia Slavica* 5, no. 343:119–124.
- Harel, David. 2001. *Rzecz o istocie informatyki. Algorytmika*. Warszawa: WNT.
- Karttunen, L., J-P. Chanod, G. Grefenstette, and A. Schiller. 1996. “Regular Expressions for Language Engineering.” *Natural Language Engineering* 2 (4): 305–328.
- Karttunen, Lauri. 1995. “The Replace Operator.” *Proceedings of ACL-95*. 16–23. <http://www2.parc.com/istl/members/karttune/publications/acl-95/acl95.pdf>.
- Karttunen, Lauri, and Kenneth R. Beesley. 2001. “A Short History of Two-Level Morphology.” ESSLLI-2001 Special Event.
- Long, Thomas Hill, ed. 1979. *Longman Dictionary of English Idioms*. Harlow and London: Longman Group Limited.
- Nunberg, Geoffrey, Ivan A. Sag, and Thomas Wasow. 1994. “Idioms.” *Language* 70 (3): 491–538. <http://lingo.stanford.edu/sag/papers/idioms.pdf>.
- Piotrowski, Tadeusz. 1999. “Tagging and Conversion of a Bilingual Dictionary for XeLDA, a Xerox Computer Assisted Translation System.” *Papers in Computational Lexicography COMPLEX '99 Proceedings*. Budapest: Hungarian Academy of Sciences, 113–120.
- Piotrowski, Tadeusz, and Zygmunt Saloni, eds. 1997. *Nowy słownik angielsko-polski i polsko-angielski*. Warszawa: Wilga.
- Prószéky, Gábor, and András Földes. 2005. “An Intelligent Context-Sensitive Dictionary: A Polish-English Comprehension Tool.” *Proceedings of L&T 2005 — Human Language Technologies as a Challenge for Computer Science and Linguistics*. Poznań, 386–389.
- Prószéky, Gábor, and Balázs Kis. 2002. “Context-Sensitive Electronic Dictionaries.” *Proceedings of COLING-02*. Taipei. <http://acl.ldc.upenn.edu/C/C02/C02-2015.pdf>.
- Przepiórkowski, Adam, Zygmunt Krynicki, Łukasz Dębowski, Marcin Woliński, Daniel Janus, and Piotr Bański. 2004. “A Search Tool for Corpora with Positional Tagsets and Ambiguities.” *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004*. 1235–1238. <http://nlp.ipipan.waw.pl/~adamp/Papers/2004-lrec/fcqp.pdf>.
- Reagan, Robert Timothy. 1987. “The Syntax of English Idioms: Can the Dog Be Put On?” *Journal of Psycholinguistic Research* 16 (5): 417–441.
- Segond, Frédérique, and Elisabeth Breidt. 1995. “IDAREX: Formal Description of German and French Multi-word Expressions with Finite State Technology.” Technical Report MLTT-022, Rank Xerox Research Centre, Grenoble.

- Segond, Frédérique, and Pasi Tapanainen. 1995. "Using a Finite-State Based Formalism to Identify and Generate Multiword Expressions." Technical Report MLTT-019, Rank Xerox Research Centre, Grenoble.
- Smith, Logan Pearsall. 1943. *Words and Idioms*. London: Constable & Company.
- Villavicencio, Aline, Timothy Baldwin, and Benjamin Waldron. 2004. "A Multilingual Database of Idioms." *Proceedings of the Fourth International Conference on Language Resources and Evaluation*. 1127–30. <http://lingo.stanford.edu/pubs/tbaldwin/lrec2004-idiomDB.pdf>.
- Zgusta, Ladislav. 1971. *Manual of Lexicography*. Praha: Academia.