An Approach to Efficient Processing of Multi-Word Units

Cvetana Krstev, Ivan Obradović, Ranka Stanković, Duško Vitas

An Approach to Efficient Processing of Multi-Word Units | Cvetana Krstev, Ivan Obradović, Ranka Stanković, Duško Vitas | Computational Linguistics - Applications, Studies in Computational Intelligence 458 | 2013 | | 458

10.1007/978-3-642-34399-5_6

http://dr.rgf.bg.ac.rs/s/repo/item/0000822

An Approach to Efficient Processing of Multi-Word Units

Cvetana Krstev, Ivan Obradović, Ranka Stanković, and Duško Vitas

Abstract Efficient processing of MWUs in the course of development of morphological MWU dictionaries is not easy to achieve, especially when languages with complex morphological structures are concerned, such as Serbian. Manual development of this type of dictionaries is a tedious and extremely slow process. To alleviate this problem we turned to our multipurpose software tool, dubbed LeXimir, in the production of lemmas for e-dictionaries of multi-word units. In addition to that, we developed a procedure aimed at making the production of MWU dictionary lemmas more efficient. This procedure, which strongly relies on our comprehensive e-dictionaries of Serbian simple words, was subsequently implemented as a new functionality of LeXimir. In this paper we present our approach, and offer an evaluation of the performance of the new functionality of LeXimir, and hence of our procedure, obtained through two rounds of experiments on various types of data. The paper ends with a brief discussion of some further possible applications of both the procedure and LeXimir in various language processing tasks.

Cvetana Krstev
University of Belgrade — Faculty of Philology, Studentski trg 3, 11000 Belgrade, Serbia e-mail: cvetana@matf.bg.ac.rs

Ivan Obradović
University of Belgrade — Faculty of Mining and Geology, Djušina 7, 11000 Belgrade, Serbia, e-mail: ivano@rgf.bg.ac.rs

Ranka Stanković
University of Belgrade — Faculty of Mining and Geology, Djušina 7, 11000 Belgrade, Serbia, e-mail: ranka@rgf.bg.ac.rs

Duško Vitas
University of Belgrade — Faculty of Mathematics, Studentski trg 16, 11000 Belgrade, Serbia, e-mail: vitas@rgf.bg.ac.rs

1 Introduction

Morphological electronic dictionaries of Serbian for natural language processing (NLP) are being developed for many years now. Their development follows the methodology and format (known as DELAS/DELAF) presented for French in [3]. E-dictionaries in the same format have been produced for many other languages. This format can be briefly described in the following way: in a dictionary of lemmas (DELAS) every lemma is described in full detail so that a dictionary of forms containing all necessary grammatical information (DELAF) can be generated from it, and subsequently used in various NLP tasks. Two corpus processing systems that support work with this dictionary format were developed, Unitex [13] and Nooj [20], both of which use finite-state technology as initially introduced in [5]. Serbian e-dictionaries of simple forms have reached a considerable size: they have a total of more than 127,000 lemmas [6] generating close to 4.4 million forms. Unitex distribution includes a large sample from the Serbian e-dictionary which covers a specific text, the Serbian translation of Voltaire's *Candide*.

The NLP community offered various approaches to lexical treatment of multi-word units (MWUs). Since 2003 the workshops on multi-word expressions are being regularly organized in the scope of major events — ACL, EACL, Coling or LREC — not to mention special sessions during other language technology or computational linguistics conferences.[1] On these occasions treatment of MWUs was presented from various points of view showing that significant results were achieved. However, two points need to be stressed. Although much has been done to expand research to less-resourced languages, they are still presented to lesser degree. The second point is that it seems that the identification and extraction of MWUs has attracted more attention of researchers than their lexical representation. Various approaches to lexical representation of MWUs were analyzed in detail by Savary [16].

Slavic languages are analyzed in [14] and arguments are presented why they are in general more difficult for NLP than Romance and Germanic languages, and which of their features are making them, nevertheless, more suitable for higher levels of processing, like parsing. However, for lexical representation of Serbian MWUs, less favorable features of Slavic languages predominate, most notably its rich morphology.

In order to produce a robust lexical representation of Serbian MWUs we applied two approaches. Productive classes of MWUs, like numerals and various named entities that rely on them (e.g. measurement phrases) can best be described by dictionaries in the form of finite-state transducers (FST), and a number of them were produced for Serbian as well [10]. Other contiguous MWUs that are idiosyncratic in nature, namely nouns and adjectives, have to be lexically described in a similar way as simple words. In the computational lexicography school led by Maurice Gross, the interest in MWUs and the production of morphological dictionaries of compounds has been vivid from the very beginning [4]. Following that direction, dictio-

[1] Programs and proceedings of these workshops can be found at http://multiword.sourceforge.net/.

naries of MWU lemmas (DELAC) that are provided with information enabling the production of all inflected forms (DELACF) were developed for several languages, including French [2], English [15], Greek [11], Italian [21], and Portuguese [12]. At Unitex official web site a comprehensive list of references related to the production of e-dictionaries of MWUs for these languages is given.

The lexical description of MWUs in the so-called DELAC/DELACF format in practice means that MWU lemmas have to be collected, generated, and inflected.

2 Inflection of MWUs

In order to produce a list of MWU forms in a systematic way, it is necessary to decide what the lemma of all these forms is, what are its additional features, how do its simple word constituents inflect, and what is the inflectional behavior of a MWU as a whole. One can imagine that for some languages this complex procedure can be skipped and a list of MWU forms can be produced from scratch. Serbian is, however, like all Slavic languages a highly inflectional language and such a shortcut procedure cannot be applied. We will illustrate this with two examples. The nominal MWUs *petokraka zvezda* 'five-pointed star' and *Farenhajtov stepen* 'Fahrenheit degree' consist of an adjective followed by a noun, which in Serbian is the natural order of an adjective and a noun in a MWU. However, these MWUs, together with a few more allow a reverse order as well — *zvezda petokraka* and *stepen Farenhajtov*. Both MWUs can be used in plural form. In Serbian, adjectives and nouns inflect in number and case, while adjective forms also depend on gender, definiteness, comparison, and in some cases animacy. Adjectives and nouns do not inflect freely in a MWU — the values of categories for number, case and gender have to agree. The animacy is important only for the masculine gender nouns in the accusative singular. Since the gender of *zvezda* 'star' is feminine, the animacy is of no relevance for this MWU. This is not the case for *Farenhajtov stepen* since *stepen* is masculine. To obtain the correct accusative singular form *Farenhajtov stepen* it is important to know that *stepen* is inanimate, otherwise the incorrect accusative form *Farenhajtovog stepena* would be obtained. Finally, adjectives *petokrak* 'five-pointed' and *Farenhajtov* 'belonging to Fahrenheit' have no comparative and superlative forms, so they will not be generated. Indefinite adjective forms are rarely used in compounds so they are not generated either.

This example illustrates the complexity of capturing all information about one MWU in its DELAC lemma. The most demanding part is to formulate the agreement conditions in a consistent way. A special form of inflectional transducers developed by Savary [17] and implemented in the Multiflex system answers most of these questions. The inflectional graph in Figure 1 illustrates this. A MWU serving as lemma is tokenized and its tokens become values of variables: in the case of *petokraka zvezda* values of variables become \$1=*petokraka*, \$2=<space>, \$3=*zvezda* while in the case of *Farenhajtov stepen* they are \$1=*Farenhajtov*, \$2=<space>, \$3=*stepen*. If a simple pattern of the form $<\$i>$ appears in the inflectional graph it means

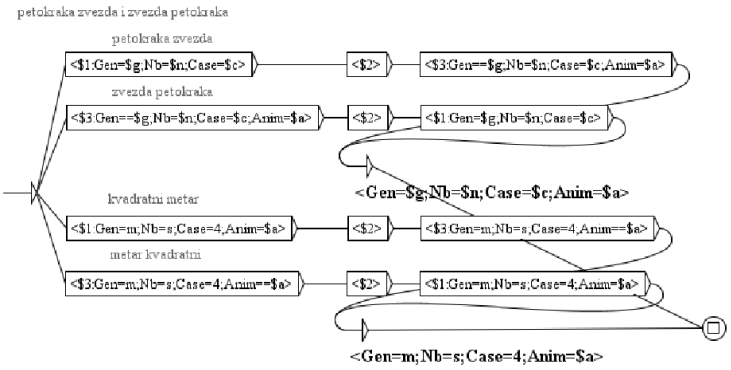

Fig. 1 A simplified transducer for compounds of the type *petokraka zvezda* and *Farenhajtov stepen*

that the corresponding token is recopied in all MWU inflectional forms as it is — in our examples the second token, a space, is reproduced in all inflectional forms.

A token pattern can be followed by one or more equations of the type *Grammatical_feature=value*. In that case the specific form of a token is needed. In our example the token <$3:Gen=m;Nb=s;Case=4> from the lower part of the graph means that the masculine gender, singular and accusative form of the third token is needed. However, the gender of the noun *zvezda* from the MWU *petokraka zvezda* is feminine, so this form cannot be produced and the lower paths in the graph will be ignored. They will not be ignored for some other MWUs, like *Farenhajtov stepen*, since the gender of *stepen* is masculine.

Additionally, grammatical-feature equations can contain not only concrete values but also unification variables. A unification variable instantiates to all values of the corresponding grammatical feature. For Serbian, a pattern <$3:Case=$c> means that forms for all cases — 7 different values — will be generated for the third token. The occurrence of the same unification variables in the same path means that their values have to agree. If a pattern <$1:Case=$c> appears in the same path as <$3:Case=$c> it means that when the genitive form of the first token is generated then the genitive form of the third token has to be generated as well, and that will also be the value of the 'Case' feature of the generated MWU form — the output of the transducer.

Finally, a unification variable does not need to instantiate to all values of some grammatical feature. Instead, it can inherit its value from a token itself. In the pattern <$3:Gen==$g> the variable $g inherits its value from the third token. For *petokraka zvezda* the variable $3 will instantiate to the value *f* since the gender of the third token is feminine, while for *Farenhajtov stepen* it will instantiate to the value *m* — the gender of the token *stepen*. In both cases, the variable $g from the pattern <$1:Gen=$g> occurring in the same path will have to agree with the value inherited from the third token; hence, in the first case it will have the value *f* and in the second case the value *m*.

The two possible orders of the adjective and the noun in the MWU are achieved with two separate paths in the graph, one for the order given by a lemma itself, and the other for the reverse order. The orthographic variants of MWUs, e.g. the optional use of a hyphen, as well as omission of some of its constituents can also be easily described using Multiflex graphs [18]. The Multiflex system is incorporated into Unitex, but it was also successfully used for Polish proper names in another environment [19]. For the inflection of Serbian MWUs 104 such transducers were developed — 18 for adjectives and 86 for nouns.

By analogy with entries in a dictionary of simple word lemmas, an entry in a DELAC dictionary consists of a MWU lemma to which a name of an inflectional transducer (similar to the one represented in Figure 1) is assigned. Similarity ends here, because simple word constituents of a MWU lemma also have to be described in a way that enables the production of all needed forms. This leads finally to the following lemma forms:

```
petokraka(petokrak.A6:aefs1g) zvezda(zvezda.N600:fs1q),NC_AXNr
Farenhajtov(Farenhajtov.A1:akms1g) stepen(stepen.N5:ms1q),
   NC_AXNr
```

These DELAC entries enable the production of all MWU forms for DELACF dictionary of forms; forms representing the genitive singular with reverse order of constituents for these two MWUs are:

```
zvezde petokrake,petokraka zvezda.N:fs2q
stepena Farenhajtovog,Farenhajtov stepen.N:ms2q
```

Production of a lemma in the format presented is far too demanding to be done manually because for each MWU one has to provide the following information:

1. What is the lemma? One has to decide that *petokraka zvezda* and *Farenhajtov stepen* are more preferable as lemmas then *zvezda petokraka* and *stepen Farenhajtov*.
2. How does this MWU inflect and which inflectional transducer should be used for it? The two example MWUs have an adjective/noun structure and allow a reverse order of constituents, therefore inflectional transducer NC_AXNr should be used.
3. Which MWU constituents inflect? In our example both constituents inflect which means that inflectional information about them is needed as well.
4. What are DELAS entries of these MWU constituents that enable the generation of all needed forms? These entries for *petokraka zvezda* are `petokrak.A6` and `zvezda.N600`, and for *Farenhajtov stepen* they are `Farenhajtov.A1` and `stepen.N5`.
5. What are the values of grammatical features of constituent forms used in the MWU lemma? For the first example they are `aefs1g` and `fs1q`, while for the second they are `akms1g` and `ms1q`.

A fully manual production of MWU lemmas is, however, not necessary, because possible answers to the above questions that concern MWU constituents can be found in dictionaries of simple words.

3 LeXimir as a Dictionary Management System

Bearing in mind the aforementioned complexity of production of MWU lemmas we have endeavored towards a procedure for automatic production of DELAC entries. The software tool which enabled the implementation of this procedure was LeXimir,[2] a multipurpose tool developed by the University of Belgrade Language Technology Group [9] to support computational linguists in developing, maintaining and exploiting e-dictionaries. LeXimir is written in C#, and operates on the .NET platform. It can run on any personal computer under Windows and supports simultaneous manipulation of various language resources: e-dictionaries, wordnets, and aligned texts.

Implementation of LeXimir followed a modular approach. Namely, there exists a common core of the system, which is coupled with several modules performing different tasks. The central part of the system is *LeXimir_Core* composed of several .Net libraries: *CommonRes.dll*, *NlpQuery.dll*, *VisualTMX.dll* and *WNDictAuto.dll* (Fig. 2). For communication with lexical resources LeXimir makes use of the *NlpQuery.dll* module. Modular organization of components provides two obvious benefits. In the first place, it enables the use of various resources in any part of the system, wherever they are needed. Thus, for example, morphological dictionaries can be used for adding additional morphological information to wordnet synsets, whereas both morphological dictionaries and the wordnet can be used in production of concordances for aligned texts. On the other hand, it enables the use of *LeXimir_Core* in different scenarios: as a standalone Windows application LeXimir.exe or as a web application *VebRanka.aspx*[3], also known as VebRanka (previously WS4QE), which is supported by the *wsQueryExpand.asm* web service. The web service accepts and generates data sets in XML format, which are further converted into data structures that can be used for different purposes (string, array, table, etc.). As examples of web service functions we will mention a few characteristic ones: *getObliciLeme(lema)*, which generates inflected forms for a given lemma, *getSinonimiWN_WithFlex(lema)*, which returns all synonyms from a given wordnet synset in all inflected forms, and *getSinonimiWN_NoFlex(lema)* which returns synonyms without inflected forms.

As our e-dictionaries are Unitex-based, and Unitex is open source software distributed under the LGPL license, we incorporated its modules in LeXimir for the majority of tasks that involve manipulation of e-dictionaries. For the production of MWU DELAC lemmas we used the appropriate Unitex modules for dictionary look-up.

LeXimir provides for concurrent manipulation of several dictionaries of lemmas, both of simple words and MWUs (DELAC), distributed in any number of files. However, the possibility of manipulating dictionaries of word forms is not envisaged, as such files are produced automatically either from DELAS or DELAC by

[2] LeXimir is available under CC_BY-NC license. For more information see http://korpus.matf.bg.ac.rs/soft/LeXimir.html

[3] http://hlt.rgf.bg.ac.rs/VebRanka

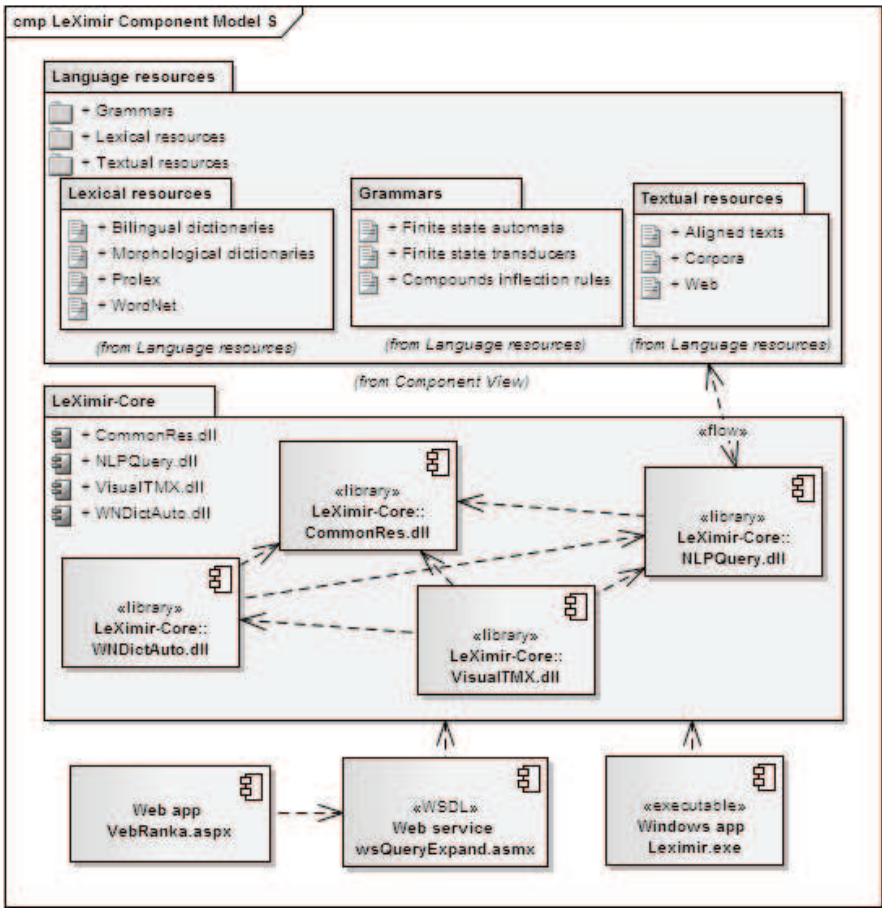

Fig. 2 Components of the software tool LeXimir

means of appropriate FSTs. Organizing dictionaries in sets of different files is practically motivated. Namely, smaller size files are much easier to manipulate.

With LeXimir's editor for MWUs the user can perform, beside the usual functions — add, insert, copy, change — many more demanding activities. The users can check the correctness of every lemma with the function 'Inflect' that lists all inflected forms of a selected lemma. Another useful function is the extraction of subsets of lemmas based on different criteria: lemmas' beginning, their part of speech (PoS), inflectional class code, syntactic and/or semantic markers or a Boolean combination of these criteria.

Figure 3 shows the table for manual production of a DELAC entry having two constituents: *petokraka* and *zvezda*. A user can insert constituents of a MWU in the column 'Form' of the table. In the next step columns 'Lemma', 'FST' (PoS and inflectional codes of constituents), and 'GramCat' (grammatical codes of constituents)

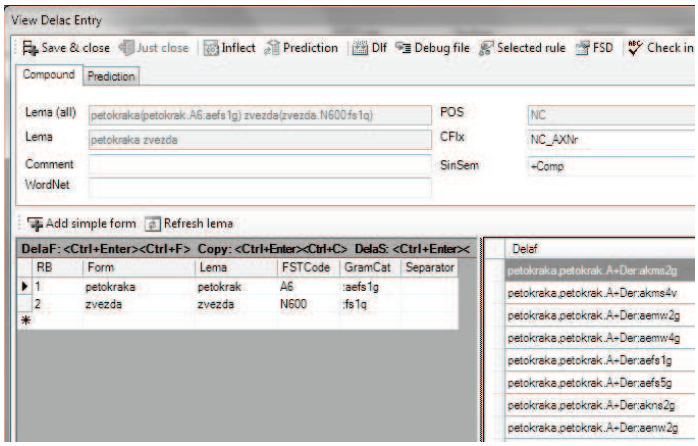

Fig. 3 The DELAC entry management form of Leximir

have to be filled. The system does this automatically by offering all possible solutions retrieved from DELAS/DELAF dictionaries of simple words. In the third step, the selection of the correct lemma, FST code and grammatical categories is supported by possible combinations offered in auxiliary tables (in the bottom right corner of Figure 3). In the final step, the user has to fill manually the code of the inflectional transducer for the newly produced MWU lemma, and attach to it the appropriate semantic and other markers. A user can then check the correctness of the new MWU lemma by using the 'Inflect' function that invokes Multiflex to perform the inflection.

The outlined procedure does help in answering the two last questions posed at the end of Section 2. However, answers to questions 2 and 3 have to be provided by the user. Thus, by following this approach not more than 2,800 DELAC entries were produced during three years, which we found very ineffective.

4 A Rule Based Procedure for Inflection of MWUs

4.1 Detection of inflectional properties of MWU lemmas

We have further improved the procedure for production of MWU lemmas when we realized that the answers obtained automatically in support of manual production of MWU lemmas can also help in detection of the syntactic composition of a MWU and therefore indicate the appropriate inflectional transducer. Namely, the MWUs in Serbian have predictable basic structures. For instance, nominal MWUs with two constituents (beside a separator) fall into five basic structures:

- Adjective/noun (both inflect and agree in gender, number and case)

- Noun/noun (both inflect and agree in number and case)
- Noun/noun in the genitive or in the instrumental (only the first noun inflects)
- Word/noun (only the second noun inflects; the first word is usually not a Serbian simple word)
- Noun/adjective (both inflect and agree in gender, number and case)

However, there are 25 different inflectional graphs for the nominal MWUs with two constituents because there are subtleties that have to be taken into consideration besides these basic structures, e.g. can a MWU have plural forms, can a separator be omitted or replaced by another separator, etc. The basic structure, however, determines the general form of a MWU lemma and information that has to be supplied for its constituents.

Table 1 Different interpretations of the sequence *živa rana*

form	lemma	translation	PoS	relevant grammatical categories
živa	*živ*	'alive'	A	nominative, singular, feminine
živa	*živa*	'mercury'	N	nominative, singular, feminine
rana	*rana*	'wound'	N	nominative, singular, feminine genitive, plural, feminine
rana	*ran*	'early'	A	nominative, singular, feminine

Thus, automatic production of the lemma for *petokraka zvezda* could proceed like this: a look-up in the dictionary of simple word forms determines that *zvezda* can only represent two realizations of the noun lemma *zvezda*, namely in the nominative singular or in the genitive plural. Similarly, it is determined that *petokraka* can be one of 12 different representations of the adjective *petokrak*; however, only one of them agrees with the noun *zvezda*, and that is the singular, feminine gender, nominative case form. Consequently, it can be deduced that only the basic structure adjective/noun applies here.

Of course, not all MWUs are so easy to process. For instance, for the MWU *živa rana* 'open wound' a dictionary look-up offers several possibilities (Table 1). Thus there are five possible MWU structures: adjective/noun, noun/noun, noun/noun in the genitive, noun/adjective, and adjective/adjective whereas only the first one is correct.

Table 2 Five lemmas offered for the sequence *živa rana*

First constituent	Second constituent	MWU inflectional class
živa(živ.A15:aefs1g)	rana(rana.N600:fs1q)	NC_AXN
živa(živa.N600:fs1q)	rana(rana.N600:fs1q)	NC_NXN
živa(živa.N600:fs1q)	rana	NC_N2X
živa(živa.N600:fs1q)	rana(ran.A17:aefs1g)	NC_NXAr
živa(živ.A15:aefs1g)	rana(ran.A17:aefs1g)	AC_AXA

Table 3 Super-class AXN

Class	Example	Translation	Specifics
AXN	*živa rana*	'open wound'	
AXN3	*Pitagorina teorema*	'Pythagorean theorem'	does not inflect in number
AXNF	*serijski ubica*	'serial killer'	second constituent changes gender in plural forms
AXNr	*petokraka zvezda*	'five-pointed star'	allows reverse order

Based on an analysis illustrated by previous examples, we have developed a new functionality within LeXimir that offers one or more DELAC entries for every MWU presented in its lemma form. As indicated by the example, it relies on information in e-dictionaries of simple words, but also uses a set of manually produced rules to deduce the basic structure of a given MWU, as well as its additional features. For the example *živa rana* this functionality would offer five lemmas; the first one would be selected, the remaining four discarded (Table 2).

In order to design our automated procedure we grouped all inflectional transducers into equivalence classes or super-classes: a super-class consists of all MWUs having the same basic structure. It also means that the form of their MWU lemma is the same because they need the same information for the production of inflectional forms. This is also reflected in the convention we used for naming the inflectional transducers: A stands for an adjective constituent, N stands for a noun constituent, X stands for a constituent that does not inflect (including a separator), with some additional digits and letters added to differentiate transducers. This is illustrated in Table 3 by four classes (names of inflectional transducers) all belonging to the same AXN super-class and used for the inflection of MWUs consisting of an adjective followed by a noun, where both constituents inflect and must agree in basic grammatical categories.

One super-class need not consist of MWUs having the same syntactic structure. For instance, a super-class N4X consists of three component MWUs for which the first component is a noun that inflects and two remaining components do not inflect. According to our DELAC dictionary MWUs belonging to this super-class may have various syntactic structures, as presented in Table 4.

Also, MWUs having the same syntactic structure need not all belong to the same super-class. Such is the case for MWUs with the syntactic structure noun/noun in the genitive. The plural forms of such MWUs, in the case that they exist, can be:

- Only the first component inflects in number, the second component does not inflect. Examples are *profesor matematike/profesori matematike* 'professor(s) of mathematics' and *red vožnje/redovi vožnje* 'travel schedule(s)';
- Both components have to be in the plural form, e.g. *teme ugla/temena uglova* 'angle vertex/angle vertices';
- The second component can be either in the singular form or in the plural form, for instance, *predsednici države/predsednici država* 'presidents of the state/presidents of states';

Table 4 Super-class N4X

Example	Translation	Structure
kola hitne pomoći	'first aid car'	noun/adjective in gen./noun in gen.
uskrsenje sina božjeg	'resurrection of the Son of God'	noun/noun in gen./adjective in gen.
menadžment ljudskim resursima	'human resources management'	noun/adjective in instr./noun in instr.
raketa zemlja-vazduh	'air-to-ground missile'	noun/noun in nom./noun in nom.
ugovor o zakupu	'lease contract'	noun/preposition/noun
trgovac na malo	'wholesaler'	noun/preposition/adjective

- both components can be in the singular and the plural form in all possible combinations, e.g. *analiza dokumenta/analiza dokumenata/analize dokumenta/analize dokumenata* 'document(s) analysis/document(s) analyses'.

Only the MWUs belonging to the first listed group belong to the super-class N2X and they require inflectional information only for the first component. All the other MWUs belong to the super-class NXNg and for them inflectional information is necessary for both components, as illustrated by the examples:

```
profesor(profesor.N2:ms1v) matematike,NC_N2X
teme(teme.N324:ns1q) ugla(ugao.N115a:ms2q),NC_N2X4
```

In order to formulate a strategy for the production of MWU lemmas we analyzed the data available in the existing DELAC dictionary looking for useful information. On the one hand, we identified the additional information assigned to components of MWUs belonging to a particular inflectional class, and on the other, we identified inflectional classes associated with the same additional information.

4.2 The rule design strategy

The procedure for automatic construction of a DELAC type dictionary relies on a manually produced set of rules. The rule design strategy resulted from the aforementioned expert analysis of available MWU lemmas. The task of the rule based procedure is to automatically generate the complete MWU lemma. However, the strategy and the procedure are independent, and changes in the strategy, in general, do not affect the procedure itself. This approach enabled us to experiment with various rule strategies, and thus the final strategy used is a result of several iterations.

Each rule consists of one set of general conditions (tags `<RuleGenCond>`) and zero to many sets of special conditions (tags `<RuleSpecCond>`). Special conditions are added to general conditions in the processing phase and one such complete set has to be satisfied in full in order to produce a possible solution — a MWU lemma. In that respect each rule behaves as a disjunction of conjunctions. For instance, the rule in Example 1 is applied to two component MWUs as follows: if com-

ponents satisfy (according to the dictionary of simple words) the specified grammatical conditions, namely, that the first is an adjective in the nominative case and the second component is a noun in the nominative case as well, and these two components agree in gender and animacy, then the additional conditions are checked, and at least one of them needs to be satisfied. In this case it means that one of the following additional conditions must be satisfied: the first component starts with uppercase letter (e.g. *Pariska komuna* 'The Paris Commune'), or both components are already in plural (e.g. *lokalni izbori* 'local elections'), or the second component is a collective noun (e.g. *kandirano voće* 'candied fruit').

Example 1 (XML form of a rule for the class NC_AXN3, super-class NC_AXN — for adjective/noun MWUs that do not inflect in number).

```
<Rule ID="2" CFLX="NC_AXN3" CflxGroup="NC_AXN">
    <RuleGenCond>
        <Word ID="1" POS="A" Flex="true" Case="1" Anim="$a"
              Gen="$g"/>
        <Word ID="2" POS="N" Flex="true" Case="1" Anim="=$a"
              Gen="=$g"/>
    </RuleGenCond>
    <RuleSpecCond ID="1" Example="Pariska komuna">
        <Word ID="1" Num="s" Cond="$PRE"/>
        <Word ID="2" Num="s"/>
    </RuleSpecCond>
    <RuleSpecCond ID="2" Example="lokalni izbori">
        <Word ID="1" Case="1" Num="p"/>
        <Word ID="2" Case="1" Num="p"/>
    </RuleSpecCond>
    <RuleSpecCond ID="3" Example="kandirano voce">
        <Word ID="1" Case="1" Num="s"/>
        <Word ID="2" Case="1" Num="s"
              SinSem="+VN,+Coll,+HumColl"/>
    </RuleSpecCond>
</Rule>
```

Another rule that applies to three-component MWU adjectives in the form of a simple word adjective followed by the conjunction *kao*, followed by an animate noun, is given in Example 2. An example is the adjective *gladan kao vuk* 'hungry as a wolf'. Adjectives of this type have two plural forms: the noun component can be either in the singular *gladni kao vuk* or in the plural *gladni kao vuci*. This rule has no additional conditions and has no agreement requests.

Example 2 (A rule for the class AC_A3XN2, super-class AC_A3XN).

```
<Rule ID="153" CFLX="AC_A3XN2" CflxGroup="AC_A3XN">
    <RuleGenCond Example="gladan kao vuk">
        <Word ID="1" POS="A" Flex="true" Case="1" Num="s"
              Gen="m"/>
        <Word ID="2" POS="MOT" Flex="false" Cond="=,kao"/>
        <Word ID="3" POS="N,A" Flex="true" Case="1"
              Num="s" Anim="v"/>
    </RuleGenCond>
</Rule>
```

Each rule can check orthographic properties of a processed MWU and/or match its components with applied dictionaries of simple words. Orthographic conditions check separators used between words (a space is presumed by default) and capitalization of components — due to the condition `Cond="$PRE"` in the first set of special conditions in Example 1 this rule is applied only if the first component is written with initial upper-case. Rules can also check whether a component matches a string, e.g. the condition `Cond="=,kao"` in the Example 2 requires that the second component of a MWU is the string *kao* (a conjunction 'as'). The other condition `Sufix="ska,ška,čka"` (Example 5) requires that the suffix of the first component is *-ska*, *-ška* or *-čka* (a comma is used as a disjunction operator).

More interesting are conditions that rely on dictionaries of simple words, and they can offer answers to following questions:

- Does a component exist in dictionaries of simple forms? For instance, due to the condition `POS="!SDIC"` in the set of general conditions in Example 5 this rule applies only if the first MWU component is not in the dictionary of simple forms (it is an "unknown word").
- What are the values of grammatical categories of a MWU component? For instance, the rule in Example 2 applies only if, according to applied dictionaries, the first word is an adjective (`POS="A"`), in the nominative case (`Case="1"`), in the singular (`Num="s"`), and in the masculine gender (`Gen="m"`).
- Do values of a grammatical category agree for two or more components? The rules use unification variables in a similar way as inflectional transducers for MWUs (described in Section 2). For instance, in Example 1 `$g` is one such variable: it receives the value of the gender from the second component (a noun) and has to agree in gender with the first component (an adjective).
- Does a component possess a specific syntactic or semantic feature? In Example 1 the third set of special conditions is applied if the second component is a collective noun or a verbal noun (`SinSem="+VN,+Coll,+HumColl"`).

In general, conditions can be negated by using two different operators: `!` and `~`. The simplest is the condition `!SDIC`, which means that a MWU component does not exist in applied dictionaries of simple words.[4]. The operator `!` is used for atomic values — for instance, the condition `Sep="!-"` requires that a component is NOT followed by a hyphen in a MWU. More often, it is used for agreement conditions. In Example 3 due to the condition `Gen="!$g"` the rule is accepted only if a MWU consists of two nouns having different gender — in our example *leptir* 'butterfly' is masculine and *kravata* 'tie' is feminine.

Example 3 (A rule for the class NC_2XN1, super-class NC_2XN — for two nouns separated by a hyphen and having different gender; the first noun will not inflect).

```
<RuleGenCond Example="leptir-kravata/bow tie">
  <Word ID="1" POS="N" Flex="false" Case="1" Gen="!$g" Num="s"
        Sep="-"/>
  <Word ID="2" POS="N" Flex="true" Case="1" Gen="=$g" Num="s"/>
</RuleGenCond>
```

[4] Similar notation is used in Unitex for meta-symbols.

The operator ˜ is the negation of the existence operator, meaning that the subset of word forms from applied dictionaries that satisfy other conditions must not contain the element satisfying a given condition. In Example 4 a rule is given that is used for MWUs in which the first component does not inflect. If the separator is not a hyphen this usually happens if the first component is not in applied dictionaries of forms, or is a prefix or an abbreviation. However, the condition `Case="˜1"` allows that the first component can also be a noun if it is NOT in the nominative case. Thus abbreviations — like *TEI* in our example — will not be rejected, although *TEI* is a homograph of a dative form of a personal name *Tea*. In this way, some cases of false ambiguity can be resolved.

Example 4 (A rule for the class NC_2XN, super-class NC_2XN — the first component does not inflect; it can be a noun, but not in the nominative case).

```
<Rule ID="17" CFLX="NC_2XN" CflxGroup="NC_2XN">
    <RuleGenCond>
        <Word ID="1" POS="MOT" Flex="false" Sep="!-"/>
        <Word ID="2" POS="N" Flex="true" Case="1" Num="s"/>
    </RuleGenCond>...
    <RuleSpecCond ID="3" Example="TEI zaglavlje/TEI header">
         <Word ID="1" POS="N" Case="˜1"/>
         <Word ID="2"/>
    </RuleSpecCond>...
```

In some rules `set` attributes of a special kind appear. They do not set conditions but rather values for the MWU lemma being generated. That is, instead of obtaining values from applied dictionaries of simple words, they allow rules to set these values themselves. They are thus used for components that do not exist in applied dictionaries ("unknown words"). In Example 5 the first component does not exist in dictionaries (`POS="!SDIC"`), but if it ends with *-ska*, *-ška* or *-čka* it will be treated as an adjective (`setPOS="A"`), with specific grammatical values (`setGramCats="np1gae"`), and a lemma, which can be obtained from the component form by deleting its final character and replacing it with an *i* (`setLemma="[B]i"`).

Example 5 (A rule for the class NC_AXN3, super-class NC_AXN — for proper names for which the first component, a relational adjective, is not in dictionaries of simple words).

```
<Rule ID="14c" CFLX="NC_AXN3" CflxGroup="NC_AXN">
    <RuleGenCond>
        <Word ID="1" POS="!SDIC" Flex="true" Cond="$PRE"
              setPOS="A" setFlexCode="A2"/>
        <Word ID="2" POS="N" Flex="true" Case="1" Num="p"/>
    </RuleGenCond>
    <RuleSpecCond ID="1" Example="Lofotska ostrva">
        <Word ID="1" Sufix="ska,ška,čka" setLemma="[B]i"
              setGramCats="np1gae" />
        <Word ID="2" Gen="n" />
    </RuleSpecCond>...
</Rule>
```

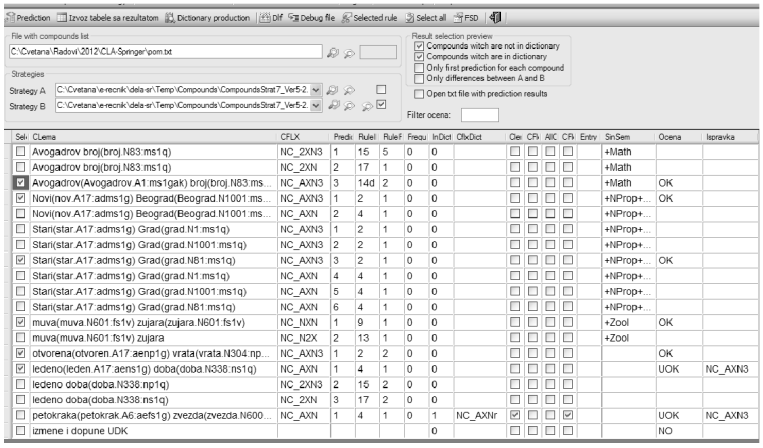

Fig. 4 Implementation of the Strategy on the prepared list of MWUs

Our rule based strategy presently consists of 117 rules — 97 for nouns and 20 for adjectives. Among them, 38 rules pertain to MWUs with 2 components, 45 rules to MWUs with 3 components, 20 rules to MWUs with 4 components, 9 rules to MWUs with 5 components, and 5 rules to MWUs with 6 and more components.

4.3 Software implementation

To manipulate the strategy in the form of a XML document our tool LeXimir relies on W3C standard languages Xquery and XSLT supported by .Net. The user interface for automatic production of DELAC lemmas is very straightforward and easy to use. A user can choose a file with a prepared list of MWUs and a file with a strategy, and the results will be presented to him in the form of a table (see Figure 4) in which the user has only to check the correct solutions upon which a list of DELAC entries is produced.

Figure 4 depicts the resulting table for a list of 8 MWUs. The third option offered by the strategy for the first MWU, *Avogadrov broj* 'Avogadro's number' is the correct solution. It was produced by a rule similar to one presented in Example 5 because the possessive adjective *Avogadrov* is not included in the Serbian DELAS dictionary of adjectives. As for the second MWU, *Novi Beograd* 'New Belgrade (a municipality of Belgrade)', the first of the two options offered by the strategy is the correct solution. For the third MWU, *Stari Grad* 'Old City (a municipality of Belgrade)' the strategy offers as much as 6 options, among which the third represents the correct solution. Such a large number of options offered is due to the fact that the form *grad* can represent as much as three lemmas: city, degree, and hail. Out of the two options offered by the strategy for the fourth MWU, *muva zujara* 'blow fly', the first one is the correct one. As for the 5^{th} MWU *otvorena vrata* 'open door

(a meeting of parents with teachers)' only one solution is offered and it is the correct one. Three possible solutions are offered for the 6^{th} MWU, *ledeno doba* 'ice age', and one of them, the first, AXN, is partly correct. Namely, the super-class is properly determined, and hence the lemma form, and what remains is to replace the inflection transducer by AXN3, as this MWU does not have a plural. The correction can be made by the user by stating the new, correct name of the transducer in the last column of this partly correct solution. The 7^{th} MWU, *petokraka zvezda* is already in the dictionary which is evidenced by the fact that the column 'ClfxDic', and the following four columns are already filled. The solution offered by the strategy is almost the same as the one existing in the dictionary, except for the fact that the strategy failed to identify that this MWU allows a reversed order of components, which is a highly exceptional feature. The option of the user interface to detect MWUs already in the dictionary is very useful, as it prevents the introduction of duplicates in the dictionary. In addition to that, it may alert the user as to the potential shortcomings of the strategy. For the 8^{th} MWU, *izmene i dopune UDK* 'amendments to UDC' no solution is offered — the MWU has an unusual structure for which no prediction was made.

When all options offered by the strategy are reviewed and those for which entries for a DELAC dictionary are to be produced ticked, the system will generate them automatically. Thus, we obtain an automated answer to questions 2 and 3 posed at the end of Section 2. Question 1 is answered by the user, who prepares the list of input lemmas. In some rare cases all rules will fail and a solution — compound lemma — will not be offered to the user. In that cases the user will have to produce a lemma consulting the existing e-dictionary, as illustrated in Figure 3.

There are various debugging tools and preference selections at user's disposal. In the strategy development phase the user can compare the results obtained by the use of various strategies on the same MWU input list. The user may also filter the results and obtain only those that differ from the results obtained by the previous version of the strategy. He/she can preview the log file to see which rules were used for a particular MWU and in which order. The user can also see which simple word forms were retrieved from e-dictionaries of simple words and what were their grammatical values.

LeXimir has been successfully used for languages other than Serbian and English, namely, for Bulgarian [8]. The new functionality for production of DELAC entries is also expected to perform successfully without any modifications for other languages. The prerequisites are that there exists a Unitex module for that language including: a dictionary of simple words in DELAS format, transducers for the inflection of simple words, the automatically produced dictionary of simple word forms DELAF, and transducers for the inflection of MWUs. As mentioned before, most of these conditions are satisfied for many languages. However, in order to apply this functionality to a new language it would be necessary to develop a new language-dependent strategy, that is, a new XML document. It is also worth mentioning that the system can be easily modified to work with formats of simple words dictionaries other than those supported by Unitex. To that end, only the dictionary look-up module would have to be changed.

4.4 Procedure Evaluation

In order to evaluate the performance of LeXimir's functionality for automated generation of MWU lemmas we have conducted experiments on two occasions. The first evaluation took place in the first phase of the development of our procedure and strategy, involving three data sets. The first set consisted of nouns and adjectives already available in the existing DELAC dictionaries. The MWU lemmas for dictionary entries were (re)produced by LeXimir and then compared to the (correct) dictionary lemmas. The second set of data consisted of common MWUs compiled from several sources, all of them nouns, while the third set consisted of a list of geographic names. In all cases the results produced by the system were validated manually.

In line with the possibility of a "partly" correct solution that we have recognized in the previous subsections, the evaluation results were classified as follows:

1. The system produced the correct lemma and assigned the correct inflectional class for a given MWU, and thus the overall solution is considered as correct;
2. The system produced the correct lemma but failed to assign the correct inflectional class, whereas the assigned super-class was correct, and thus the overall solution was considered as partly correct;
3. The system offered one or more solutions, but they were all rejected as incorrect;
4. The system failed to offer a solution.

The results of the first evaluation showed that for the first set of data our system produced 73.42% of correct results for noun MWUs and 77.07% for adjective nouns (88.14% and 97.07% respectively if we take into account the partly correct solutions), for the second data set consisting of nouns 85.92% of correct results (96.39% with partly correct), and for the third set of geographic names 57.92% of correct results (61.39% with partly correct). Hence, the results varied substantially depending on the type of data used. These results are discussed in more detail in [7].

In the meantime we have used our system intensively, amended it and refined our strategy. Then, we have conducted a second round of evaluation using three new data sets. The first two contained MWUs from a terminological dictionary for library and information sciences (LIS): the first data set included 519 MWUs of a more general nature, which are used outside this restricted domain, whereas the second set included 1,114 MWUs belonging to specific library and information science terminology. In addition to that, we used a smaller set of 152 MWU proper names, mostly geographic names and event names.

As in the case of the first evaluation the results varied depending on the type of data used: for the first data set of general terms the system produced 84.97% of correct results (96.14% if we include partly correct solutions), for the second data set of specific terms 78.01% of correct results (88.42% with partly correct), and for the third set of geographic names 93.42% of correct results (there were no partly correct results). However, when looking at these results, one must also take into account that the size of data sets also varies considerably. Although a comparison with the results obtained in the first evaluation would seem natural, it is not easy to

draw a conclusion whether we have made a substantial improvement in our strategy from the first evaluation cycle, given the relative heterogeneity of the type and size of data involved. It should, however, be noted that specific terms from the second LIS dictionary data set are often artificial, due to the nature of controlled dictionaries, and thus tend to be longer than average MWUs and consequently closer to free phrases. Hence, we will refrain from a general conclusion and just point out that in the case of relatively comparable sets of geographic names from the first evaluation and proper names from the second, a considerable improvement was reached beyond doubt.

In the second evaluation we also looked at the relation between the number of MWU components and the results obtained. As it was to be expected, the percentage of correct results decreases with the size of the MWU: very close for MWUs with two and three components (83.75% and 83.73% respectively, or 94.45% and 91.12% with partly correct) it drops to 70.08% (83.46% with partly correct) for MWUs with four components, 64.29% with five, and only 17.65% for MWUs with six components. There were no partly correct solutions in the last two cases. As for the one MWU with seven components, two with eight and one with nine that appeared in our data sets, the system was unable to offer a solution at all.

Although the system in some cases offered as much as eight possible solutions for a single MWU, the correct one, if it existed, was always within the first five, most often the first. This also depended on the size of the MWU, namely, for MWUs with two components the first option offered was the correct solution in 86.8% of cases and for MWUs with three components in 95.78% of cases. For MWUs with four, five and six components, if a correct solution was found, it was always the first one offered, although only in rare cases a second option was even offered.

In general, it is safe to say that the results obtained in both evaluation cycles testify to the fact that our approach yielded a strategy and procedure which can greatly contribute to efficient processing of MWUs.

5 Existing and Further Applications

The outlined procedure is now in everyday use for the production of MWU dictionary entries for Serbian. Due to the new functionality implemented in LeXimir the size of the MWU dictionary grew from the initial 2,800 lemmas to existing 9,600 in a relatively short period. We expect this growth rate to be even greater in the forthcoming period, as many new MWU lists are being prepared.

The benefits obtained by including the MWU dictionary in language processing tasks for Serbian are already clearly visible. Besides the benefits that were to be expected, it has been already shown that the MWU dictionary can also be very useful in text disambiguation [1], and further in the parsing process [22]. We would like to point out another interesting aspect of MWUs which can be exploited in the processing of named entities, as the initial phase in information extraction. Serbian morphological dictionaries and local grammars are successfully being used for

recognition of names of persons and of various functions they might perform within the society [10]. Local grammars for recognition of functions can recognize various syntactic structures but, naturally, not all of them. The use of MWUs can contribute to the increase of the recall without further complicating the local grammars. For example, the local grammar does not recognize the function of the person acting as *specijalni izaslanik UN za pregovore o statusu Kosova Marti Ahtisari* 'UN special envoy for negotiations on the status of Kosovo Martti Ahtisaari' because the addition *o statusu* 'on the status' is not foreseen by the local grammar. When *pregovori o statusu* 'negotiations on the status' are added to the MWU dictionary, the local grammar covers the aforementioned structure as well. This example leads us to possible applications related to inflection of free noun phrases based on the recognition of their syntactic structure (as shown by successful processing of specific LIS terms in pervious section).

This approach has already been tested in VebRanka [9]. Namely, as the described procedure for production of DELAC entries was implemented in the core engine of LeXimir it can be used not only in all parts of LeXimir but also in VebRanka, which as we have seen, was in a way built "on top" of LeXimir. This enables expansion of queries submitted to the Google search engine. The main feature of VebRanka is that it enables inflection of simple words, MWUs and free phrases supplied as keywords to Google. The tool relies on Serbian e-dictionaries, inflection transducers for simple words and MWUs, and uses Unitex and Multiflex modules for inflection and dictionary look-up. As for the free phrases that are not in the MWU dictionary, VebRanka relies on its built-in strategy, and always chooses the first of the options offered, which is, as we have seen, the correct one in most cases.

Query expansion in the web environment offers different levels for expansion details. VebRanka accepts the query from the user and submits it to the local web service, which then expands the query and forwards it to the Google search engine. To that end the Google AJAX Search API is used, a Java script library which provides for embedding Google searches into web pages or web applications. The abundance of Google services (Web Search, Local Search, Video Search, Blog Search, News Search and Book Search) are used by this library, consisting of simple web objects aimed at performing "inline" search.

Acknowledgements This research was supported by the Serbian Ministry of Education and Science under the grant #III 47003.

References

1. Alegria, I., Ansa, O., Artola, X., Ezeiza, N., Nojenola, K., Urizar, R.: Representation and Treatment of Multiword Expressions in Basque. In: Second ACL Workshop on Multiword Expressions: Integrating Processing, pp. 48–55. Barcelona, Spain (2004)
2. Courtois, B., Garrigues, M., Gross, G., Gross, M., Jung, R., Mathieu-Colas, M., Silberztein, M., Vivs, R.: Dictionnaire électronique des noms composés delac : les composants NA et NN. Rapport Technique 55, LADL, Paris, Université Paris 7 (1997)

3. Courtois, B., Silberztein, M.: Dictionnaires électroniques du français. Larousse, Paris (1990)
4. Gross, M.: Lexicon-grammar. the representation of compound words. In: Proceedings of Coling 1986, pp. 1–6 (1986)
5. Gross, M.: The use of finite automata in the lexical representation of natural language. In: Electronic dictionaries and automata in computational linguistics, *Lecture Notes in Computer Science*, vol. 377, pp. 34–50. Springer (1989)
6. Krstev, C.: Processing of Serbian — Automata, Texts and Electronic Dictionaries. Faculty of Philology, University of Belgrade, Belgrade (2008)
7. Krstev, C., Stanković, R., Obradović, I., Vitas, D., Utvić, M.: Automatic Construction of a Morphological Dictionary of Multi-Word Units. In: IceTAL, pp. 226–237. Springer, Reykavik, Iceland (2010)
8. Krstev, C., Stanković, R., Vitas, D., Koeva, S.: E-Connecting Balkan Languages. In: Proc. of the Workshop on Multilingual Resources, Technologies and Evaluation for Central and Eastern European Languages — RANLP09, pp. 23–29. Borovetz, Bulgaria (2009)
9. Krstev, C., Stanković, R., Vitas, D., Obradović, I.: The Usage of Various Lexical Resources and Tools to Improve the Performance of Web Search Engines. In: 6th LREC. Marrakech, Marocco (2008)
10. Krstev, C., Vitas, D., Obradović, I., Utvić, M.: E-dictionaries and finite-state automata for the recognition of named entities. In: Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing, pp. 48–56. Association for Computational Linguistics, Blois, France (2011). URL http://www.aclweb.org/anthology/W11-4407
11. Kyriacopoulou, T., Mrabti, S., Yannacopoulou, A.: Le dictionnaire électronique des noms composés en grec moderne. Lingvisticae Investigationes (2002)
12. Mota, C., Carvalho, P., Ranchhod, E.: Multiword lexical acquisition and dictionary formalization. In: Proceedings of the Workshop Enhancing and Using Electronic Dictionaries, Coling'2004, pp. 73–77. Geneva, Switzerland (2004)
13. Paumier, S.: Unitex 2.1 User Manual. http://www-igm.univ-mlv.fr/unitex/UnitexManual2.1.pdf (2011)
14. Przepiórkowski, A.: Slavonic information extraction and partial parsing. In: Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies, ACL '07, pp. 1–10. Association for Computational Linguistics, Stroudsburg, PA, USA (2007). URL http://dl.acm.org/citation.cfm?id=1567545.1567547
15. Savary, A.: Recensement et description des mots composés - méthodes et applications. Ph.D. thesis, Université de Marne-la-Vallée (2000)
16. Savary, A.: Computational Inflection of Multi-Word Units — A Contrastive Study of Lexical Approaches. Linguistic Issues in Language Technologies **1**(2) (2008)
17. Savary, A.: Multiflex: A Multilingual Finite-state Tool for Multi-Word Units. In: CIAA, pp. 237–240 (2009)
18. Savary, A., Krstev, C., Vitas, D.: Inflectional Non-compositionality and Variation of Compounds in French, Polish and Serbian, and Their Automatic Processing. Bulag — Bulletin de Linguistique Appliquée et Générale **32**, 73–94 (2007)
19. Savary, A., Rabiega-Wisniewska, J., Wolinski, M.: Inflection of Polish Multi-Word Proper Names with Morfeusz and Multiflex. In: Aspects of Natural Language Processing, *Lecture Notes in Computer Science*, vol. 5070, pp. 111–141. Springer (2009)
20. Silberztein, M.: Nooj: A Linguistic Annotation System for Corpus Processing. In: Proceedings of HLT/EMNLP on Interactive Demonstrations, HLT-Demo '05, pp. 10–11 (2005)
21. Vietri, S., Elia, A., D'Agostino, E.: Lexicongrammar, electronic dictionaries and local grammars in italian. Lingvisticae Investigationes (2004)
22. Wehrli, E., Seretan, V., Nerima, L.: Sentence Analysis and Collocation Identification. In: Proc. of the Multiword Expressions: From Theory to Applications — MWE 2010, pp. 28–36. Beijing, China (2010)