



MONDILEX:

**Conceptual Modelling of Networking of
Centres for High-Quality Research in Slavic
Lexicography and Their Digital Resources**

**Department of Knowledge Technologies
Jožef Stefan Institute**

Research Infrastructure for Digital Lexicography

Information Society – IS 2009

MONDILEX Fifth Open Workshop

Ljubljana, Slovenia, 14 – 15 October, 2009

Proceedings

Ljubljana 2009



MONDILEX: Conceptual Modelling of Networking of Centres for High-Quality Research in Slavic Lexicography and Their Digital Resources

Department of Knowledge Technologies, Jožef Stefan Institute

Research Infrastructure for Digital Lexicography

Information Society 2009

MONDILEX Fifth Open Workshop Ljubljana, Slovenia, October 14 - 15, 2009

Proceedings

Tomaž Erjavec (Ed.)

The workshop is organized by the project

GA 211938 MONDILEX

*Conceptual Modelling of Networking of Centres for High-Quality
Research in Slavic Lexicography and Their Digital Resources*

supported by EU FP7 programme Capacities – Research Infrastructures

Design studies for research infrastructures in all S&T fields

Research Infrastructure for Digital Lexicography

Ljubljana, Jožef Stefan Institute, 2009.

The volume contains contributions presented at the Fifth open workshop “Research Infrastructure for Digital Lexicograph”, held in Ljubljana, Slovenia, on October 14 - 15, 2009. The workshop is organized by the international project GA 211938 MONDILEX *Conceptual Modelling of Networking of Centres for High-Quality Research in Slavic Lexicography and Their Digital Resources*, Capacities – Research Infrastructures (Design studies for research infrastructures in all S&T fields) EU FP7 programme.

Workshop Programme Committee

Tomaž Erjavec (Chairperson)

Jožef Stefan Institute, Ljubljana, Slovenia

Ludmila Dimitrova (Co-chairperson)

Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria

Radovan Garabík

Ľ. Štúr Institute of Linguistics, Slovak Academy of Sciences, Bratislava, Slovakia

Leonid Iomdin

Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia

Violetta Koseska-Toszewa

Institute of Slavic Studies, Polish Academy of Sciences, Warsaw, Poland

Volodymyr Shyrovok

Ukrainian Lingua-Information Fund, National Academy of Sciences of Ukraine, Kyiv, Ukraine

Workshop Organisation

Tina Anžič

Jožef Stefan Institute, Ljubljana, Slovenia
and the organisers of the Information Society '09 meta-conference

Editor of the volume: **Tomaž Erjavec**

© Editors, authors of the papers,

Jožef Stefan Institute 2009

Foreword

This volume contains articles presented at the Fifth open workshop of the MONDILEX project “Research Infrastructure for Digital Lexicography”. The workshop was organised by the international project GA 211938 MONDILEX *Conceptual Modelling of Networking of Centres for High-Quality Research in Slavic Lexicography and Their Digital Resources, Capacities – Research Infrastructures*, developed under EU FP7 programme. The workshop, organised by the Jožef Stefan Institute in the scope of the Information Society - IS 2009 metaconference, was held on 14–15 October 2009 in Ljubljana, Slovenia.

The main purpose of this workshop is to study and outline innovative solutions the conceptual design for new research infrastructures, esp. on a European scale. It addresses two tasks: the architecture and functional characteristics of MONDILEX’s Knowledge Grid and the conceptual scheme for a research infrastructure of networking of centres for high-quality research in Slavic lexicography and their digital resources.

We hope the workshop results will be useful to lexicographers, computer linguists and linguists in general.

Ludmila Dimitrova, Tomaž Erjavec

Contents

Structure Editor: a Powerful Environment for Tagged Corpora	1
<i>Leonid Iomdin, Victor Sizov</i>	
Empowering Human Language Technologies with Grid.....	13
<i>Jan Jona Javoršek, Tomaž Erjavec</i>	
Slovak Paremiography Database.....	20
<i>Peter Ďurčo, Radovan Garabík</i>	
The Japanese-Slovene Dictionary jaSlo: a Usability Study	27
<i>Kristina Hmeljak Sangawa, Tomaž Erjavec</i>	
Integrating the Polish Language into the MULTTEXT-East Family: Morphosyntactic Specifications, Converter, Lexicon and Corpus.....	37
<i>Natalia Kotsyba, Adam Radziszewski, Ivan Derzhanski</i>	
Adding Multi-Word Expressions to sloWNet.	56
<i>Špela Vintar, Darja Fišer</i>	
The Digitisation and Deployment of the Slovenian Biographical Lexicon.....	64
<i>Jan Jona Javoršek, Tomaž Erjavec, Petra Vide Ogrin</i>	
Bulgarian-Polish-Lithuanian Corpus – Problems of Development and Annotation	72
<i>Ludmila Dimitrova, Violetta Koseska, Danuta Roszko, Roman Roszko</i>	
Future and Possibility.....	87
<i>Violetta Koseska1 and Antoni Mazurkiewicz</i>	
Theory of Lexicographic Systems. Part 3	98
<i>Volodymyr Shyrovok</i>	
Using Ukrainian National Linguistic Corpus in Lexicography.....	120
<i>Oleg Bugakov</i>	

Structure Editor: a Powerful Environment for Tagged Corpora^{*}

Leonid Iomdin, Victor Sizov

A.A.Kharkevich Institute for Information Transmission Problems,
Russian Academy of Sciences

iomdin@iitp.ru, sizov@iitp.ru

Abstract. The paper describes a software environment, Structure Editor, or StrEd, which is used to create and maintain a deeply tagged corpus of Russian texts. StrEd is language-independent and may be used in sophisticated corpora compilation for any language. Special attention is given to Intellectual Debugger, a module of StrEd that provides a technology for versatile checking and improvement of syntactic annotation of the corpus.

1. Introductory Remarks

SYNTAGRUS, a deeply tagged corpus of Russian texts, is the first Russian Treebank that offers, in addition to morphological and lexical annotation, full syntactic annotation for each sentence in the form of dependency tree structures, as well as partial lexical functional annotation (Apresjan et al. 2006, Nivre-Boguslavsky-Iomdin 2008). The corpus is part of the ETAP-3 multipurpose linguistic processor developed by the Laboratory of Computational Linguistics of the Kharkevich Institute (see e.g. *Apresjan et al.* 2003a). Deeply annotated corpora, including SYNTAGRUS, are used in diverse research paradigms and a large number of applications, such as information extraction and retrieval, parsing evaluation, learning phases of statistically-driven parsers etc.

The paper describes the language of corpus annotation and outlines tools and techniques used to develop and maintain SYNTAGRUS, which constitute the multipurpose environment called Structure Editor.

2. Syntactic Markup Language

In order for a text corpus to be functional in a broad range of applications, it must meet a number of important criteria. In particular, 1) it must feature several layers of linguistic data that can be extracted from the annotation independently of each other; and 2) it should be scalable and incrementable both quantitatively and qualitatively so that new types of information could be added easily.

Naturally, such a corpus requires a data representation language that allows the implementation of such features. In addition, it must be supplied by standard programming means for text parsing, sophisticated search, and conversion. XML meets all of these conditions and has therefore been chosen as the corpus markup language.

At present, the following XML elements are used for markup.

The whole document (corresponding to one text in the corpus) is referred to with the **TEXT** tagging element, which has a **ver** attribute (tagging format version). **TEXT** contains two further elements: 1) **info**, which contains identification data about the text concerned, and 2) **body**, which lists all sentences of the text.

The **info** element has a number of attributes, including the **title** (full name of the text), **source** (book, newspaper, journal, website etc), **author** (one or more authors of the text), **date** (the file's last-modified

^{*} This research has been funded in part by a grant from the Russian Foundation of Basic Research (No. 07-06-00339) and the Program of the Section of historical and philological sciences of the Russian Academy of Sciences "The text in interaction with the sociocultural environment: levels of historical literature and linguistic interpretation", for which the authors are grateful.

time), **annot** (the name(s) of the linguist expert(s) who correct the tagging).

In any text file, sentences are specified by **S** tags: the initial and final tags <S> and </S> define the sentence borders. Sentence properties are presented with three attributes: **ID** (sentence number), **Comment** (experts' informal comments), and **Status** (the state of the sentence during the tagging, which may be assigned by the parser or the expert who corrects parsing results and, roughly, shows the correctness degree of the syntactic structure).

Each word of the sentence, including punctuation marks that may be assigned to it¹, are marked by the **W** tag. The initial and the final tags define the borders of the string of characters corresponding to the word form. Morpho-syntactic information on the word is presented with the following four attributes: **ID** (word number in the sentence); **Lemma** (dictionary entry form), **Feat** (the set of morphological features), **Link** (incoming syntactic relation, i.e. the syntactic relation going from the syntactic parent into the word); **Dom** (number of the word which is the syntactic parent of the word under consideration; if this word is the syntactic head of the sentence, the value of Dom attribute is **_root**).

The newest version of SYN_{TAG}RUS contains partial lexical functional annotation: for collocations that could be presented with the apparatus of lexical functions (LF, see e.g. Apresjan *et al.* 2003b, 2007) the tagging includes information on values and attributes of such lexical functions. Lexical functional annotation is presented with **LF** tagging elements, one element for each occurrence of the lexical function attribute/value pair. The element has three attributes: **LFFUNC** (name of the lexical function); **LFARG** (number of the word in the sentence that serves as the attribute of this lexical function) and **LFVAL** (number of the word in the sentence that serves as the value of this lexical function).

Since the corpus was annotated using the main parser and a few other modules of the ETAP-3 linguistic processor, the annotation includes certain data specific for ETAP: such data are necessary when the corpus is used as the benchmark resource for ETAP performance evaluation, or a regression test resource. On the sentence level, two parameters are used: the **PARAMS** attribute is used (the list of text processing parameters set up by the ETAP parser when processing this particular sentence) and the **TRANS** attribute (which offers the English equivalent of the sentence concerned as produced by ETAP-3 MT module). On the word level, three more attributes are used: **KNAME** (the name of the entry of the combinatorial dictionary corresponding to the word of the sentence) and **EXTRAFEATS** (a list of auxiliary ETAP features assigned to the word, such as the zero feature assigned to a noun or an adjective if these serve as predicate in the absence of a copula verb), and **HYPOT** (which stores the information on the particular parsing rule that established the syntactic link going into the word from its parent).

In addition to the above attributes, there are additional attributes for storing auxiliary information that facilitates the process of corpus annotation. In particular, the **EXTRACOMM** attribute presents information on annotation errors for individual words and for the whole sentence. Another attribute, **SRCFILE**, which stores the information on text title and sentence identification, was introduced for the sake of the technology of error correction offered by the Structure Editor environment: this technology enables one to create a separate file containing erroneously parsed sentences from different texts, correct the parses, and substitute the improved sentences for the old ones directly in the original files. One more attribute, **CLASS**, is used by annotators to manually classify sentences of the corpus according to any features that the annotators may consider relevant.

¹ Most of the punctuation marks are considered to be assigned to the word that precedes them (no matter how many punctuation marks occur one after another). Exceptions are left brackets and left quotation marks, as well as dashes opening the sentence, which are assigned to the word following them.

Fig. 1 below offers an example of morpho-syntactic annotation with the markup language for the Russian sentence *Петр крепко спит* ‘Peter is fast asleep’.

```
<S ID="1" >
<W DOM="3" EXTRAFEAT="CAP" FEAT="S ЕД МУЖ ИМ ОД" ID="1" KSNAME="ПЕТР"
ЛЕММА="ПЕТР" LINK="предик"> Петр</W>
<W DOM="3" FEAT="ADV" ID="2" KSNAME="КРЕПКО" ЛЕММА="КРЕПКО"
LINK="обст">крепко</W>
<W DOM="_root" EXTRAFEAT="ЛИЧ"
FEAT="V НЕСОВ НЕПРОШ ИЗЪЯВ 3-Л ЕД" ID="3" KSNAME="СПАТЬ"
ЛЕММА="СПАТЬ">спит</W> </S>.
```

Fig. 1. XML annotation of a Russian sentence

The EXTRAFEAT attribute “CAP” in the first line shows that the first word *Петр* is capitalized; the same kind of attribute “ЛИЧ” in the sixth line marks the finite form of the verb *спит* (this feature does not come from the morphology but assigned at a latter stage).

3. Annotation Tools

Considering the significant size of SYNTAGRUS (which currently has over 500,000 words and constantly growing) the annotation process has to be automated to the fullest extent possible. On the other hand, automatic annotation has to allow for verification and, if need be, correction by a human expert. In particular, the need for eventual correction ensues from high morphological, lexical and syntactic ambiguity of sentences, some of which can only be resolved using extralinguistic information, which cannot always be expected from automatic parsers. This means that the environment has to provide for comfortable viewing and editing of annotated texts. Besides, it turns out that linguist annotators often overlook the errors made by the automatic parser and themselves make a considerable amount of mistakes while editing (the human factor). This is why the environment has to provide tools for easy detection and correction of mistakes.

In order to illustrate the operation of the system of annotation tools, let us consider the annotation of the following text fragment:

Возлюбленную его звали Маргаритой Николаевной. Все, что Мастер говорил о ней, было сущей правдой. Он описал свою возлюбленную верно. Она была красива и умна.

(М. Булгаков, Мастер и Маргарита)

‘His beloved’s name was Margarita Nikolaevna. Everything the master told [the poor poet] about her was the exact truth. He described his beloved correctly. She was beautiful and intelligent.

(M. Bulgakov. Master and Margarita. Translated by Larissa Volokhonsky)

3.1. Preprocessing

The first step in the annotation of any text is to segment it into sentences. The tool used to solve this task is called Sentence Chopper. The segmenting algorithm is rather simple: it is based on the presence of end-sentence punctuation marks (dots, exclamation and question marks) paragraph characters, and the like, with a few rules that clear punctuation marks of sentence end marker status in contexts that are likely to be abbreviations (similar to English *i.e.*, *etc.*, *cf.* and the like), initials before or after surnames etc. No powerful statistics is used here, since, after automatic segmentation is finished, the result is submitted to a human editor, who corrects segmentation errors, typos, and other small defects of the text.² After all segmentation corrections are made, the text is stored as an XML document to be further processed by the main annotation program, Structure Editor.

² It is worth mentioning that since the creation of the tagged corpus requires significant manual labour at more complex stages, it is hardly reasonable to concentrate efforts to fully automate the preliminary stages of corpus processing.

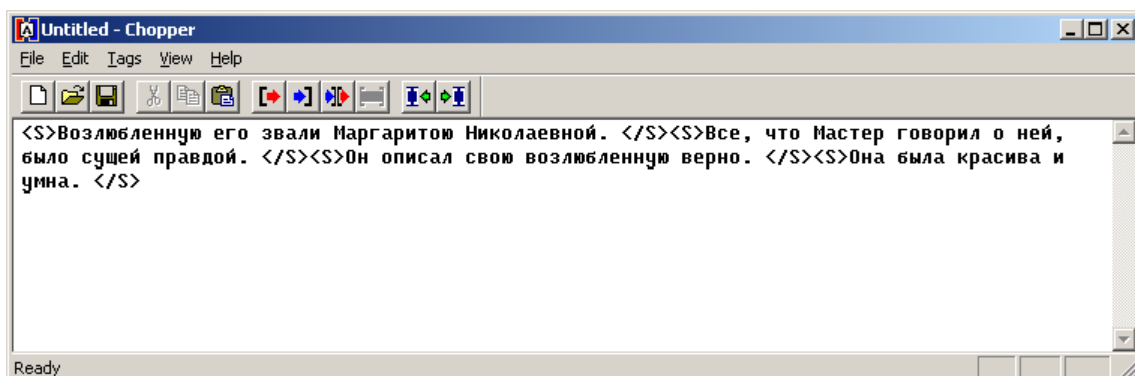


Fig.2. Screenshot of the Sentence Chopper showing the sample text segmented into sentences.

The XML document produced as the result of this segmentation is as follows:

```
<?xml version="1.0" encoding="windows-1251"?>
<text ver="1.1"><body>
<S>Возлюбленную его звали Маргаритой Николаевной. </S>
<S>Все, что Мастер говорил о ней, было сущей правдой. </S>
<S>Он описал свою возлюбленную верно. </S>
<S>Она была красива и умна. </S>
</body></text>
```

3.2. Structure Editor

Structure Editor (StrEd) is a complex software environment aimed at 1) automatic generation of morpho-syntactic and lexical functional annotation of texts, 2) manual editing of annotation results, and 3) fully manual annotation. Automatic generation is only possible for texts in natural languages that are supported by the ETAP-3 linguistic processor (at the moment, these include Russian and English³). Automatic annotation is only possible if ETAP-3 linguistic processor is installed on the computer, otherwise, only manual editing and annotation can be performed. In principle, Structure Editor is not language-specific and can be used for annotation of texts in any natural language, primarily one with rich morphology, so that the Structure Editor perfectly suits all Slavic languages.

Let us consider typical actions performed by the corpus annotator using the StrEd. If the environment is launched on a computer with ETAP-3 pre-installed, the annotator chooses the configuration of the ETAP-3 processor: the language and the database in which grammar rules and dictionaries are stored. Besides, the annotator may choose one of the viewing options for the data with which he or she is planning to work.

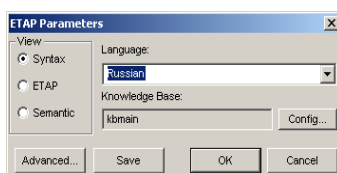


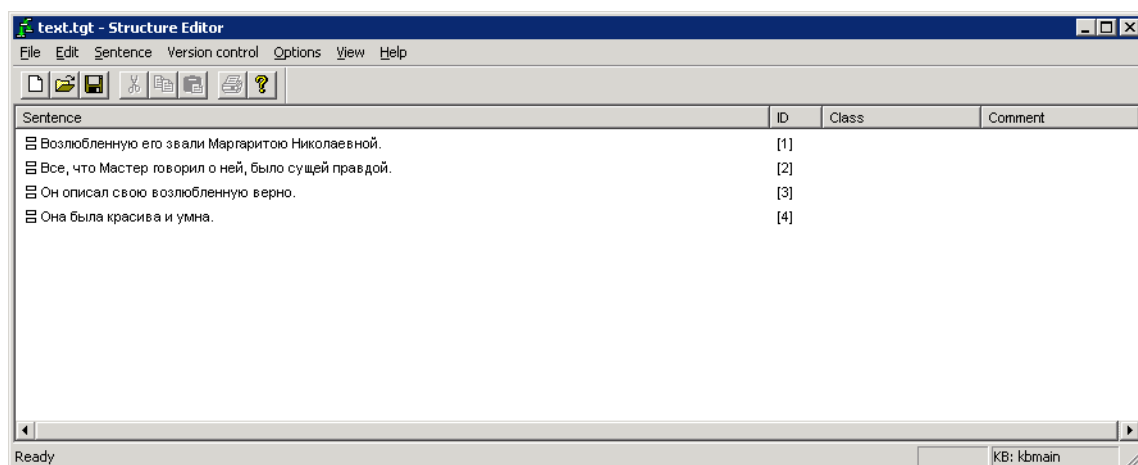
Fig. 3. ETAP-3 Initialization

StrEd allows the annotator to use diverse dialog interfaces enabling him or her 1) to view the whole text; 2) to view a sentence as a table in which every line corresponds to a particular word of the sentence; 3) to

³ Within ETAP-3, small MT prototypes are available for five more languages: French German, Spanish, Arabic and Korean; however the coverage is not sufficient to produce tagged corpora for these languages.

view the syntactic dependency tree for a sentence; 4) to view information on a particular word of the sentence; 5) to view the discrepancies within the results of automatic tagging and manual tagging of a sentence.

After the text to be processed has been loaded, the annotator receives the list of text sentences in the form of a table. Each row corresponds to a sentence. The first column of each row contains the initial text of the sentence in normal orthographic form plus a pictogram that presents the annotation status of the sentence. The latter reflects the degree to which the annotation is ready: 1) it may be fully tagged; 2) the tagging may contain lemmas and grammatical features but no syntactic tree and no lexical functional annotation, or, at an early stage, 3) it can just show word forms and punctuation marks. The screenshot in Fig. 4 corresponds to the latter case.



Sentence	ID	Class	Comment
Возлюбленную его звали Маргаритою Николаевной.	[1]		
Все, что Мастер говорил о ней, было сущей правдой.	[2]		
Он описал свою возлюбленную верно.	[3]		
Она была красива и умна.	[4]		

Fig. 4. StrEd view presenting the sample text at an initial stage with no morphosyntactic tagging performed.

The next three columns present the sentence number, the class, which could be defined by the annotators according to their needs, and the informal comment regarding the sentence. The sentences may be sorted according to the values of any of the four columns in ascending or descending order.

Menu options permit the annotator to generate automatic tagging for all sentences or some of the sentences by selecting them in the table; load individual sentences for manual editing; divide the text into parts or merge several texts into one; split or merge sentences, or perform a number of other tasks.

As a rule, the first step of text annotation is automatic tagging. After it is obtained, the sentences are revised by the annotator, who detect and corrects the errors. To conveniently view the dependency tree structure and manipulate with it, Edit Structure dialog (Fig. 5) can be used.



Fig. 5. Edit Structure Dialog of StrEd.

In this view, the annotator can easily perform all typical actions that modify the original tagging; in particular, the editor can rearrange the structure or delete the syntactic relations by simple mouse gestures, alter the lemmas, syntactic links, or grammatical features.

If these operations do not suffice to obtain the desirable results, the annotator may continue the editing by switching to another dialog, intended for sentence properties viewing and manipulation, which allows performing less typical operations with the sentence (Fig. 6).

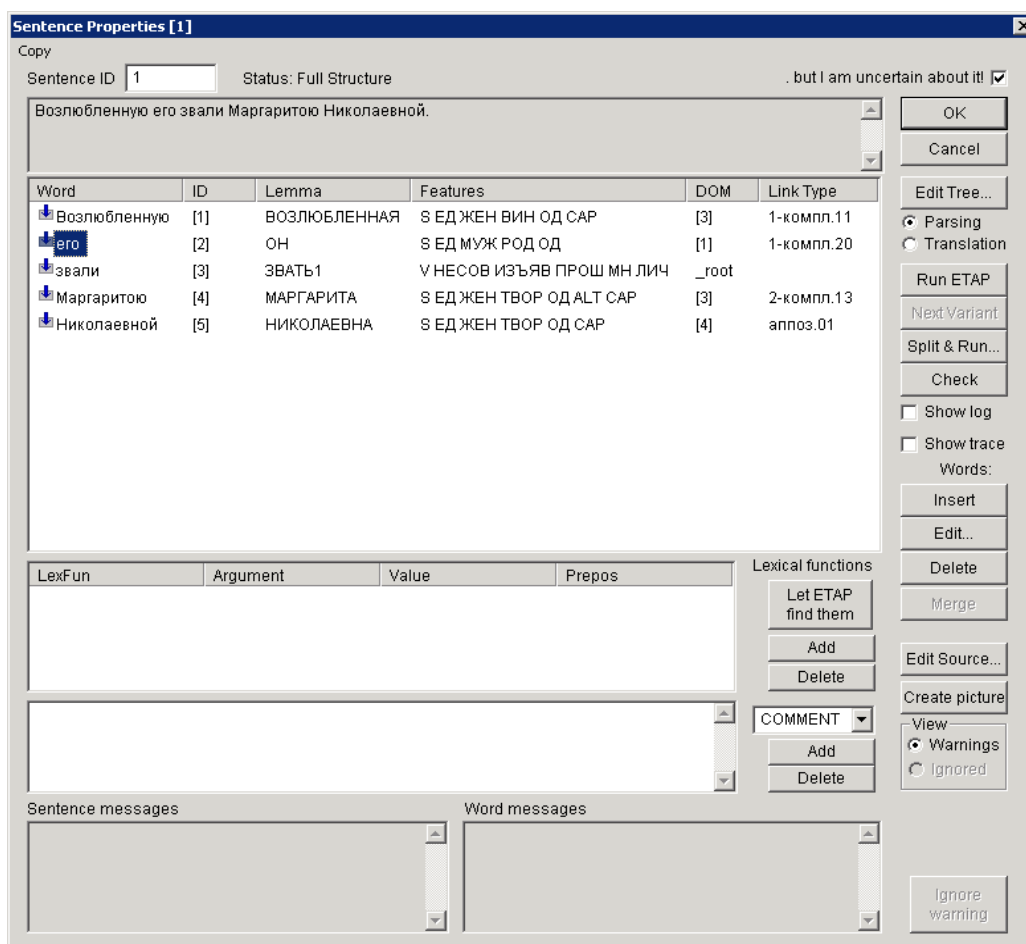


Fig. 6. Sentence Properties Dialog of StrEd.

In Fig. 6, the *Edit tree* command launches the structure editing interface described above and illustrated by Fig. 5. The *Edit Source* option opens the XML text file corresponding to the sentence processed: it may be edited manually if the annotator finds it convenient. The *Create picture* option produces a graphic file offering an image of the sentence structure as it is shown to the user in the structure editing interface.

Most of the operations that can be performed using the sentence properties dialog may be classed under five groups: 1) advanced editing of the sentence; 2) running the ETAP-3 syntactic parser on the sentence; 3) editing of lexical functional annotation; 4) processing of arbitrary tagging attributes, and 5) tackling system warnings. We will consider some of these operations in more detail.

Advanced editing allows modifying the annotation of individual words in a special dialog window, merge several neighbouring words into one, add new words into the sentence, or remove the existing ones.

Two or more adjacent words should be merged into one if these words are the result of specific processing by ETAP-3 parser of composite words like *семиголовый* ‘seven-headed’ or *Минобразования* ‘Ministry of

Education'. Since such words are naturally absent from the ETAP-3 dictionary, ETAP's morphological analyzer splits them into two (or more) words (*семи* 'seven' + *головый* 'headed' or *мин* (abbreviation of *министерство* 'ministry', explicitly mentioned in the respective entry of the morphological dictionary of ETAP-3) + *образования* ('education' in the genitive case). A similar approach is taken for the analysis of certain syntactic amalgams like *негде* ≈ 'there is no place where one could' or *некого* ≈ 'there is no one who could be', which are split into a fictitious verb with the meaning of non-existence *не* ≈ 'there is none' and the following relative pronouns. This approach is very useful for sentence processing in ETAP-3, e.g. for machine translation purposes. However, for a number of reasons (including the fact that corpus users are not necessarily people well-versed in linguistic subtleties), such units cannot be left split in the tagged corpus so they had to be merged again by hand.

The ETAP-3 parser can be run on the annotated sentence in one of two options – Parsing and Translation, which are launched by the respective radio buttons (see the rightmost part of Fig. 6). Parsing is the default mode, resorted to in order to produce the first result of syntactic parsing or alternative parses, produced by pressing the Next Variant button one or more times. The Translation mode is not used for corpus annotation; rather, it is employed to debug the machine translation options on the annotated sentences.

Large sentences (50 words or more) are best annotated using the Split & Run option. In this case, ETAP-3 parser produces syntactic structures for sentence fragments, whereupon the structures are manually merged into one. Sentence splitting is also performed by the annotator, whose expertise enables him or her to split the sentence into linguistically natural fragments, as exemplified by Fig. 7. This technique increases the chance of producing adequate parses for each of them.

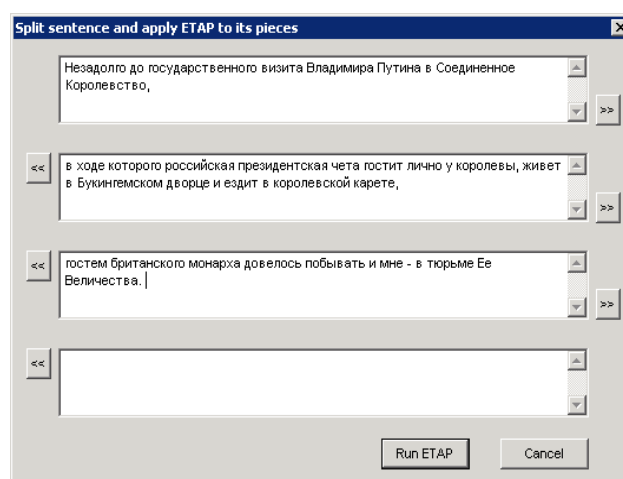


Fig. 7. Splitting long sentences for best annotation results

3.3. Lexical Functional Annotation

Lexical functional annotation of a corpus sentence can be produced in three ways: 1) automatically, together with syntactic parsing by running the ETAP-3 parser on the sentence; 2) automatically, by running a subset of ETAP-3 rules on the ready syntactic structure of the sentence approved by the expert; using the StrEd option "Let ETAP find them (LFs)", 3) manually.

The list of LF argument and values, irrespective of the way it was produced, can be manually edited: information on functions can be modified, added, or removed.

3.4. Error Detection and Correction

To demonstrate how the Structure Editor can help detect errors and correct them, let us consider a typical situation where errors emerge as a result of a human annotator's neglect.

Imagine that the annotator edits information of a particular word in the sentence using the word properties dialog box (Fig. 8). This dialog box can be opened from the sentence properties dialog (see Fig. 6 above) with a double click on the row of the table corresponding to the word in question:

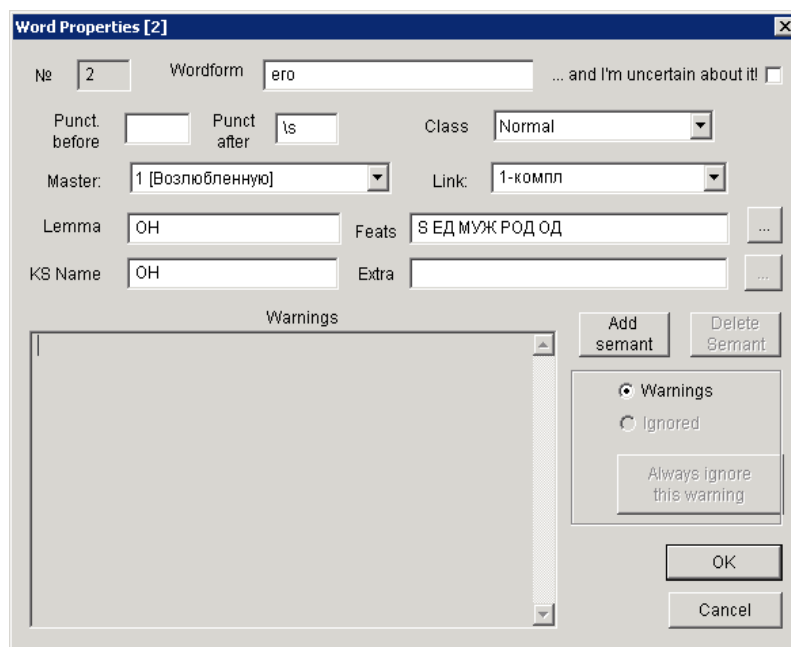


Fig. 8. Word Properties Dialog of StrEd.

The dialog box allows editing the morphosyntactic features of the word in a way similar to Edit Structure Dialog (see Fig. 5 above) but, in addition, it allows altering the word form, punctuation before or after it and viewing any diagnostic messages or warnings related to the word. The screenshot in Fig. 8 presents the second word of Sentence 1 of our illustrative text.

Imagine now that the annotator working with the word substituted by mistake the name of syntactic relation for “3-комп”, a non-existent one, or the morphological feature “од” (animate) for “одуш” (also non-existent). To detect such errors, StrEd makes use of the lists of admissible names of syntactic relations and morphological features. Such lists have to be created for each language of the corpus to be annotated. If a word is ascribed morphological features or syntactic relations outside the list, the annotator gets a warning message when trying to close the dialog window:



Fig. 9. StrEd’s warning message box

The dialog window cannot be closed unless the annotator enters a relation from the list, such as «3-КОМПЛ»⁴.

Let us now consider a slightly more complicated case, when the annotator mistakenly adds wrong features or deletes the necessary ones. Imagine that the word in Fig. 8 lost the animateness feature. To prevent such a development, StrEd uses a list of admissible strings of morphological features for different parts of speech. The features are presented with a notation similar to Backus normal form. To give an example, the

⁴ As a matter of fact, this relation is present in the list but is still erroneous in our case, since the correct name of the relation is 1-компл. We will show later how such errors can be detected.

morphological features of Russian nouns are presented in the following way:

$NOUN=S + (КОТОРЫЙ | LATIN_S | SIGNS | ((ЕД + GENDER | MH + (GENDER | \$EMPTY)) + (CASE | \$EMPTY) + ANIMATENESS) | GENDER + ANIMATENESS + CJ)$;

This means that the animateness feature should be ascribed to all nouns except for the signs \$, %, °, the word *который* ‘which’ and the words written in Roman characters. If a noun does not have an animateness feature, the module responsible for verifying the consistency of the morphological feature set will detect a discrepancy and display a warning message:

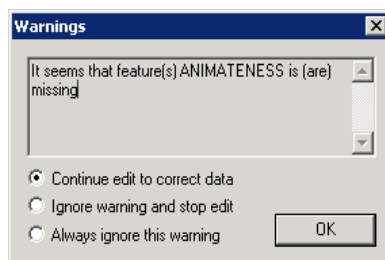


Fig. 10. Inconsistency warning message box

Unlike the previous case, such a discrepancy is not necessarily an error, since theoretically there might be exceptions not yet described in the morphology configuration file. For this reason, the annotator may ignore the warning message if he or she see it fit. The message can be ignored once, so that when the diagnostic procedure for this word is launched again, the warning reappears), or permanently. In the latter case, the warning is stored in word annotation and, should the word be subject to a new check, the diagnostic messages are compared with the stored ones and eventually ignored.

The annotator can review the list of the ignored messages for a word pressing the *Ignored* button, which appears in the Word Properties Dialog Box if there are ignored messages in its annotation. If one decides that the message should not be ignored, the list of ignored messages could be deleted.

Imagine that in the course of further tagging the expert restored the animateness of the word *ezo*, but left untouched the erroneous link 3-компл. The structure containing this link cannot be reproduced by the parser. It may seem at first glance that in such a case the annotation error could be diagnosed by producing the set of all sentence parses: if the syntactic structure offered by the annotator is absent from this set, then it is obvious that a mistake has been made. However, such an approach proved to be of little use. First, ETAP parser is not always able to produce an ideal structure, especially if the sentence is long and/or complicated, so that a false diagnosis is quite likely. Second, even if the diagnostics is correct, it may fail to determine exactly what annotation errors have been made, in which case the information that ETAP parser is unable to confirm that the verified structure is admissible becomes totally useless.

3.5. Intellectual Debugger

In order to diagnose nontrivial annotation errors, a powerful instrument, Intellectual Debugger (IntelDeb), was specially created by one of the authors (Victor Sizov) in order for the human editor to verify, in one quick step, whether the current syntactic annotation of a sentence (probably the result of several human interventions) is compatible with at least one of the parsing in principle achievable through the automatic ETAP-3 parser. As a matter of fact, IntelDeb can be considered as a specific parser which, unlike the regular ETAP parser, does not produce multiple parses of a sentence. Instead, if the IntelDeb finds that the structure being subject to verification is inadmissible, its goal is to diagnose the cause, or causes, of the situation as precisely as possible.

The Check option of the Structure Editor triggers a powerful instrument, Intellectual Debugger, specially created by one of the author’s (Victor Sizov) in order for the human editor to verify, in one quick step, whether the current syntactic annotation of a sentence (probably the result of several human interventions) is compatible with at least one of the parsing in principle achievable through the automatic ETAP-3 parser.

The underlying idea is to run the parser consecutively on all binary subtrees as presented by the annotation and see whether the existing syntactic rules and dictionaries permit the construction of such subtrees. The Intellectual Debugger algorithm checks all rules with regard to a specific syntactic link (there may be dozens of such rules and all possible lemmas for the given pair of words, starting with the rules and lemmas cited in the annotation but gradually loosening the grip and resorting to other rules and lemmas if the current choice cannot be confirmed. The debugger helps the annotator to detect errors in the annotation as well as errors in the ETAP-3 parser and/or dictionary: this could happen if an obviously correct syntactic structure produced by a human or with human intervention could not be confirmed by ETAP.

In order to better understand the principles of IntelDeb operation, let us summarize the parsing process performed by ETAP-3 parser. It consists of the following stages:

1. Morphological and pre-syntactic analysis;
2. Generation of the set of hypothetical syntactic links;
3. Verification of context conditions for the syntactic link;
4. Generation of the dependency tree.

The morphological analyzer receives a text as a sequence of character, segments it into words and produces, for each word, a set of possible morphological parses, each of which consists of a lemma name and a set of morphological features. Pre-syntactic analyzer assigns certain features to words, increases or decreases priorities of homonyms, and deletes some of the obviously irrelevant homonyms. The parser proper starts by producing all possible syntactic hypotheses, using dictionary information, morphological features, and the linear arrangement of words. As a result, an oriented graph (matrix) of syntactic links is built, whereupon every link is checked for compatibility with other links of the graph.

A link proves inappropriate and is deleted from the graph if 1) the graph does not have links or words whose presence is necessary for such a link to exist; 2) the graph has links or words whose presence contradicts the conditions of the legitimacy of the link considered and these links or words are **final** so that should such links or words be eliminated from the graph, the generation of a tree becomes impossible.

The last stage of parser operation – generation of the dependency tree – is performed according to the following algorithm:

1. The top node of the tree is chosen, or a node is linked to a fragment built around the top node; deletion of nodes and links that contradict the choice made.
2. Application of rules checking the contents and filters. Deletion of links for which conditions do not hold.
3. If the tree is built, the procedure is considered successful; if not, the operation continues; if the tree is impossible to build, the algorithm retreats one step and chooses another variant, if there are no more variants, the algorithm builds an emergency tree.

Proceeding from the above, let us consider the reasons why the parser may not confirm a given sentence parse. It may happen in the following cases:

- The morphological analyzer was unable to obtain given parses for one or more words. In especially hard cases, even text segmentation into words may be different.
- The syntactic parser could not build one or more syntactic links of the given parse.
- The given words, their features, and syntactic links were individually confirmed but further on some of them were deleted since they contradicted each other.

For this reason, to reveal the cause why a given parse cannot be confirmed becomes a hard and far from trivial task. So, on the one hand, the given link may fail to be built because the morphological analyzer did not provide the expected lemma and/or its features. On the other hand, the parse could be obtained but later deleted because a given link was absent. It must be said that the deletion process for words and links may become avalanche-like. Accordingly, in order to correctly diagnose the ultimate cause of failure, the IntelDeb must prevent the recurring deletion of words and links.

The problem of recurring deletion is solved as follows: the absence of a required word or link is observed

before the moment when it can trigger the deletion landslide: a diagnostic message is produced, whereupon the sentence is supplied with a word or link that serves as a substitute for the absent element. In case of an absent link the IntelDeb creates a special link whose name is identical with the absent one: this link cannot be deleted by further processing.

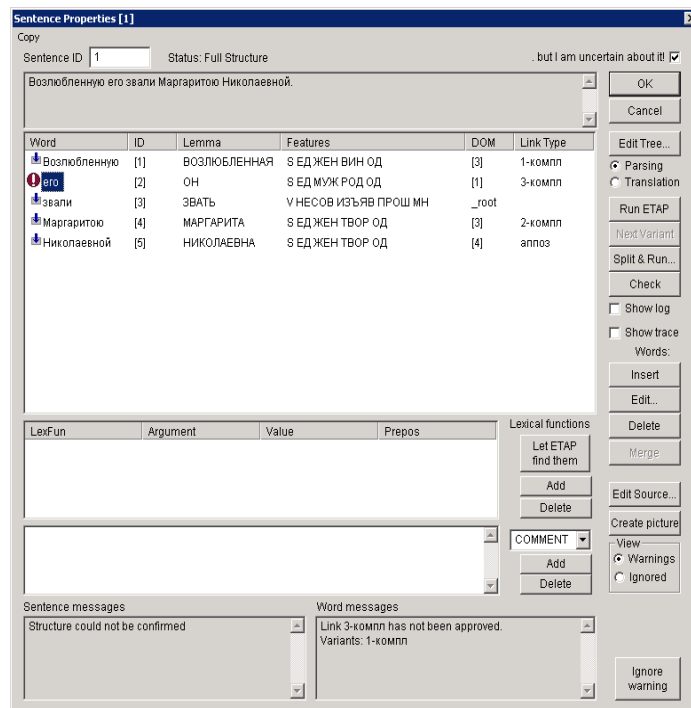
Substitution of a missing word is less trivial. If the lemma name is known, the IntelDeb loads the respective entry from the dictionary, assigning the required features. If the lemma name is unknown, a template entry is chosen, on the basis of morphological features, from the list of entries describing broad classes of unidentified words (animate feminine nouns, transitive or intransitive verbs, and the like).

We can now present the algorithm of IntelDeb operation, which consists of the following stages:

- Loading the structure to be verified and extracting the text of the sentence.
- Morphological analysis of this text.
- Checking whether a morphological parse exists for all words of the sentence. For missing parses, a diagnostic message is generated and a substitute word is chosen.
- Generating hypothetical syntactic links.
- Checking whether the required links exist for every word of the sentence. In case of a missing link, a diagnostic message is generated and a substitute link is formed. Links whose names do not coincide with the required ones are deleted..
- Launching the procedure of tree generation, checking for the required links and words at every step. If these are missing, diagnostic messages are generated and substitutes are formed.
- Launching the tracer for syntactic rules responsible for the production of the required links. If IntelDeb cannot confirm the correct structure, viewing the tracer operation step by step helps the annotator understand the causes of errors: in most cases, they are connected with errors in syntactic rules or dictionary entries.

As the result of IntelDeb processing of a tagged sentence, either the parse is confirmed, or diagnostic messages are produced which show unconfirmed morphological parsed or syntactic links. Another outcome of this processing is tracing of syntactic rules.

In our example, the diagnostic message will look as follows:



It is obvious that this algorithm allows the Intellectual Debugger to find errors in structures produced by humans or with human interference, as well as in structures produced automatically. This means that it can be used in a broad range of natural language processing tasks..

Bibliography

- [1] Apresjan Juri, Igor Boguslavsky, Leonid Iomdin, Alexander Lazursky, Vladimir Sannikov and Victor Sizov, Leonid Tsinman (2003a). ETAP-3 Linguistic Processor: a Full-Fledged NLP Implementation of the MTT // *MTT 2003, First International Conference on Meaning – Text Theory* (June 16-18 2003). Paris: Ecole Normale Supérieure, 2003. P. 279-288.
- [2] Apresjan Ju., I. Boguslavsky, L. Iomdin *et al.* Lexical Functions as a Tool of ETAP-3 (2003b). // *MTT 2003. First International Conference on Meaning-Text Theory*. Paris: Ecole Normale Supérieure, June 16–18, 2003.
- [3] Apresjan Juri, Igor Boguslavsky, Leonid Iomdin, Boris Iomdin, Andrei Sannikov and Victor Sizov (2006). A Syntactically and Semantically Tagged Corpus of Russian: State of the Art and Prospects // *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*. Genoa. P. 1378-1381.
- [4] Apresjan Ju., Igor M. Boguslavsky, Leonid L. Iomdin and Leonid L. Tsinman (2007). Lexical Functions in Actual NLP-Applications // *Leo Wanner (ed.) Selected Lexical and Grammatical Issues in the Meaning-Text Theory. In honour of Igor Mel'čuk*, Amsterdam: Benjamins Academic Publishers. ISBN 978 90 272 3094 2. P. 199-230.
- [5] Nivre, Joakim, Igor Boguslavsky, Leonid Iomdin (2008). Parsing the SYNTAGRUS Treebank of Russian // *Coling 2008. 22nd International Conference on Computational Linguistics. Proceedings of the Conference. Vol. 2*. P. 641-648. ISBN: 978-1-905593-47-7.

Empowering Human Language Technologies with Grid^{*}

Jan Jona Javoršek and Tomaž Erjavec

Jožef Stefan Institute
Jamova ulica 39, SI-1000 Ljubljana, Slovenia
jan.javorsek@ijs.si tomaz.erjavec@ijs.si

Abstract. The paper discusses a number of example uses of grid processing in digital lexicography, specifically in corpus processing and application of statistical methods to corpus data. We show how the use of parallelized grid architecture can facilitate corpus data access, corpus morphosyntactic annotation, n -gram processing and terminology extraction, and show how these methods can be applied in the context of research tools and how they can be adopted to the use in the context of web application interfaces for linguistic researchers. Finally, a work-plan to implement the infrastructure on a larger scale with multiple corpora and in different language contexts is discussed.

1 Introduction

Increasing computing requirements for acquiring and processing large textual data-sets and working with larger and larger annotated corpora in Human Language Technologies (HLT) and related disciplines represent a clearly defined problem as well as an opportunity: increasing computing resources are available and such increases in data-set and computing power put new methods and better statistical models in our hands. While the use of annotated corpora and therefore processing of large amounts of texts is only one of many types of tools used in digital lexicography (i.e. tools for working directly with lexica or machine-readable dictionaries), this paper is concerned with the domain of corpus investigations and the way a powerful and modern infrastructure for corpus development and analysis as well as development of related and derived tools can be built on the basis of the emergent grid technology and with the use of European grid infrastructure.

We have previously presented an overview of existing proposals for the use of computing grid in human language technologies (cf. Tamburini [21], Carroll et al. [3], Neuroth et al. [17], Luís et al. [11], Martins et al. [13], [14], [15], Marujo and Martins [16]) and reviewed which problems in the fields of corpus linguistics and digital lexicography map well to the use of computing grids (Erjavec and Javoršek [9]): we postulated that the use of computing grid infrastructure can benefit the fields of corpus linguistics and digital lexicography by **(1)** giving us access to sufficient computational power that would either permit us to process data-sets of a new order of magnitude or, alternatively, would permit much faster development of new tools, techniques, linguistic models and new resources, such as annotated corpora and lexicographic databases; **(2)** enabling us to use the distributed model of computing grid to solve the problem of storing, replicating and maintaining large data-sets while at the same time using its well established methods of enforcing authentication and authorization throughout the infrastructure to ensure that copyright restrictions and other limitations to the use of digital resources are always respected and enforced; and **(3)** enhance the collaborative aspect of our work with the introduction of the concept of a virtual organization (VO), which is used in the context of grid infrastructure as the entity that helps regulate the use of the infrastructure for the members of a project or a research field by providing a central place where authorization and access restrictions, but also common tools, data formats, meta-data catalogues and other resources are managed.

In the present paper, we present a number of experimental implementations of different tools on the grid that we implemented in the frame of the Mondilex project, and report on the current plans to develop a fully operational Virtual Organization for Human Language Technologies in computing grids.

^{*} The study and preparation of these results have been partly supported by the EC's Seventh Framework Programme [FP7/2007-2013] under the grant agreement 211938 MONDILEX.

2 Parallel Processing: Corpora, Tasks and Jobs

Grid computing is a form of distributed parallel computing whereby a "virtual supercomputer" is composed of loosely-coupled clusters of networked computers, acting in concert to perform very large tasks. Typically, the clusters of a computing grid are owned by different organizations, possibly working in different research fields, and located in different regions or states: their only connection is their interface to the central resource managers that keep track of the use of computing and storage resources and allocate free resources according to capabilities, policies and needs of their owners and of the users who request resources to perform useful tasks. In practice, grid infrastructure is implemented in the form of middleware, a software layer that connects a job (a specific instance of a user's program and data) to the central resource managers and computing cluster management software of a computing centre in such a way that the user is as isolated as possible from the particularities of implementation in the actual environment where the job is executed.

To enable this kind of sharing of resources, a virtual method of authentication and authorization needs to be put in place. In the case of grid infrastructure, this is implemented as a public key infrastructure (PKI) using digital certificates for authentication of users, hosts and services and the concept of virtual organizations that allow a group of users (i.e. participants in a project or members of a research community) to dictate the use of specific resources and to define what software and hardware requirements are present in the local environment of the computer where a job is going to be executed.

Grid infrastructure is in this manner able to perform computationally intensive tasks and store and share large amounts of data while permitting all participants to manage their local resources independently and ensuring a high level of data protection and controlled access, as needed in many fields that use the technology, such as medical imaging processing.

To enable grid processing, one must ensure that on all grid sites (i.e. clusters) that support a VO, a suitable software platform is available. Such a platform is called an execution environment. We have developed a small testing execution environment running inside Scientific Linux CERN 5 distribution of GNU/Linux (the upcoming standard for European grid infrastructure) and made sure that suitable software packages for our experiments were installed and functional inside it.

Secondly, some kind of remote storage has to be available for jobs to retrieve data they need to operate on and to store the results. While we are planning to develop a meta-data catalogue to manage multiple distributed storage systems, we have simplified the task by using a single storage system for the experiments.

Finally, a way to divide up the workload in several chunks and to make individual jobs process suitable chunks has to be set in place: these larger units, collections of jobs related by their purpose and data source, are called tasks. Usually, VOs develop suitable tools for creation and management of tasks and jobs. As it has been the purpose of these experiments, among other things, to find most suitable ways of managing corpus data to structure jobs and tasks, we have developed a number of scripts to handle this work automatically. A more capable system using the meta-data server and live grid resource information will have to be developed for the use of HLT VO community.

It is only natural that grid technology has been noticed as a possible match to requirements of human language technologies and its growing demand for computing power and storage, including the gains that shared but controlled storage could mean to the discipline at large. As we have already reviewed the current efforts (Erjavec and Javoršek [9]), we shall focus on the three use-cases we have implemented on the grid and on the immediate requirement to make the current grid infrastructure available to the users in the fields of digital lexicography and human language technologies in general.

3 Morphosyntactic Annotation (Tagging)

Automated annotation is a time consuming and computing intensive task, so it has been considered for our experiment. Our effort has been based on ToTaLe, an automated multilingual annotator

(Erjavec et al. [8]). Since ToTale has recently had a new tag-set added for Slovenian (Erjavec and Krek [10]), an experimental re-tagging of the FidaPLUS corpus of modern Slovenian (621 million words), seemed a natural task to do on the grid.

FidaPLUS is stored in the form of 44,000 files encoded in the Text Encoding Initiative format and contains full morpho-syntactic annotation (lemma, MSD tag) and marks for punctuation and sentence boundaries.

To perform the annotation, a new execution environment has been created on the experimental setup for the future HLT VO, and ToTale with its dependencies and language models has been installed.

In splitting up the task of annotation into a suitable number of jobs, we have aimed at targeting the maximum amount of available computing cores (680 at the time of the experiment), and for that reason we have prepared a script that created job description files containing approximately 70 files (with minor differences due to differences of file sizes), which gave us 630 jobs.

The actual job consisted of the job description file (specifying the input and output data files, execution environment, hardware requirements, start-up script etc.), a small control script and a filter that extracted the plain texts from the compressed annotated corpus files in TEI XML form (cf. Sperberg-McQueen and Burnard [20]) and passed them to ToTale in sequence, compressing the results on the fly.

The actual run has shown the mean time of execution per job to be around 10 hours, 2 hours of which have been spent queuing (waiting for computing resources) and in file upload or download. The task has been completed in under 12 hours, while consuming on the order of 6500 hours of computing time and processing and regenerating over 70 GB of corpus data—automatically annotating a 621-million words corpus in less than a day.

Practical applications of this service, particularly having in mind that ToTale supports several MULTEX-East languages and tag-sets and will, hopefully, some day support all of them (cf. Manandhar et al. [12], Erjavec [5]), are obvious to most linguistic users.

4 *n*-gram Processing

The second experimental task we executed on the grid was an example of *n*-gram statistics, in this case we collected frequencies for 1-grams and 2-grams for the whole FidaPLUS corpus separately for words, lemmas and MSDs, totaling a corpus of 1863 million words.

Due to *n*-gram counting being a much simpler task compared to automatic annotation, we have been able to ship the counting program and control script directly with the jobs (no installation in the execution environment necessary; we use Ted Pedersen's *n*-gram statistics package for Perl, Banerjee and Pedersen [1]) and could also process more files (500) per job. This resulted in 90 submitted jobs which finished in under 4 hours and consumed under 80 hours of computing time.

Again, the source files of the corpus had to be downloaded, uncompressed, processed so that relevant data was extracted from TEI XML form in a plain text file and then processed.

Since these jobs have been much shorter, clearly more time (but not computing resources) was spent queuing or downloading and uploading data than in actual processing, although it has to be noted that this occurred only in some cases (where due to faults in network transfers, files had to be downloaded several times) and most jobs finished around the second hour mark.

5 Term Extraction

Our third experiment was based on a term extractor described in Vintar ([22]) and its web-based interface. The web interface takes a text file, performs the necessary conversions (text, PDF and different office formats are accepted), uses the ToTale web service to lemmatize and annotate it and runs an *n*-gram statistical analysis on the lemmatized text. Using a combination of statistical scores based on lexical statistics and linguistic extraction (based on MSD patterns), a list of possible candidates for terminologically relevant terms in the text is generated.

We have been able to use the previously developed grid-adapted versions of both n -gram analysis and ToTaLe automated annotator to construct a fully parallel implementation of the term extraction service which can take a corpus of texts in the form of a compressed archive, split it up into several jobs and perform a sequence of conversion, annotation and n -gram counting on each file in parallel, combining the n -gram counts of all the files in the particular jobs, and finally combining the final counts in the web service when the jobs finish their work. The final task of application of statistical scores and MSD patterns is performed and results are presented directly from the web application.

Finding a predictable and efficient algorithm to split up the workload in a given number of jobs to make the process as fast as possible and to decide in which cases the grid implementation will finish faster than the direct one, however, proved more difficult. In addition, since most users submit short texts where the grid would impose an excessive penalty on the process, we could not gather a significant number of real-use examples where the grid approach could be analyzed, but if we can judge from the use of ToTaLe annotator, in time linguistic users will demand more power and submit more and more structured data, so we are confident a reliable way to perform the operation on larger corpora has not been implemented in vain.

6 From Grid to Web Services

Currently, our efforts have been concentrated on the minutiae of job and task management and grid resource allocation. While such an approach could be acceptable for researchers that want to develop new tools, researchers that want to merely use them will require more flexible and easy to use interfaces, usually in the form of web services.

As ToTaLe already has a web interface (<http://nl2.ijs.si/analyze/>), including a facility allowing a user to upload a small corpus as a compressed archive), it has been relatively easy to adapt the web application to use the grid backend to perform the annotation and to enable the service to process much larger data-sets in a reasonable time. Similarly the task for the term extractor was straightforward. Providing a web interface for a generic n -gram processing service seems less likely at this time, since the work to perform depends heavily on a number of factors, such as the structure of the corpus, the kind of n -gram analysis required etc.

For such task, we are planning to add some web-based interfaces to grid resources, possibly structured around the meta-data catalogue (further discussed in the next section). This interface should enable a user to quickly set up a number of generally useful but computationally expensive tasks, where the system should take care of factors such as the management of individual jobs, necessary conversions of corpus data and allocation of suitable grid storage for end results.

We realize that such generic services could be useful for building new services, so next to a browser interface, a programmable web API (REST and SOAP) is being considered.

7 The Future: HLT VO

In order to provide the power of grid computing to researchers in the domains of digital lexicography, corpus processing and human language technologies in general, the technology needs to be accessible as a part of dedicated grid infrastructure. Luckily, modern grid infrastructures support this approach in the form of Virtual Organizations (VOs), self-contained infrastructure elements that provide authorization management, software distribution, tools development and organizational support for a project or disciplinary community in the grid. As proposed in Erjavec and Javoršek ([9]), we are in the process of setting up a VO for Human Language Technologies: the Human Language Technologies Virtual Organization (HTL VO, <http://www.htlvo.org/>). Here we describe a number of steps that are being taken to provide this service to the community.

Creation of Core Services. To support the HLT VO, we have set-up a Virtual Organization Membership Service (VOMS) server to provide VO user and service access control. To use the server, a user (organization or person) has to get a grid digital certificate for authentication and use the

server to apply for accreditation. To support the VO, any organization can include the HLT VO VOMS configuration in its authorization control set-up, thus allowing a combination of local and VO controls to govern access to data and services of HLT VO members.

At the time of this writing, HLT VO VOMS is supported by the SiGNET cluster and it is included as a supported service in the Slovenian National Grid Initiative project. Any organization wanting to participate in the HLT VO can enroll with the VOMS to use the infrastructure and include its configuration in the local set-up to support the infrastructure locally.

Registration of the VO. While we are prepared to register HLT VO as a supported VO in the European grid infrastructure (i.e. with the EGEE and NorduGrid projects), we have not yet done so as at the time of this writing, no organizations from other nations support the VO and so it lacks international membership. We hope that with future expansions of membership, this will soon be rectified.

As soon as HLT VO is registered, it will be discoverable using the central services of both above mentioned infrastructures. It is also expected to become one of the supported VOs in the future European Grid Initiative (which is to start its operations in 2010).

As we register the VO, as members of the EGEE project, we are also planning to include support for the widely used gLite grid middleware, but as this has not been necessary for the present testing, only the easier-to-use and more efficient NorduGrid ARC has been supported.

For NorduGrid ARC, sites that already use it can start supporting the new VO simply by editing the relevant setup files and installing the software base for the job execution environment from the VO repository (cf. Ellert [4]).

Data and Metadata. Due to many restrictions that are often applied to the use of corpus data according to contracts regulating the use of copyrighted and other non-free materials, it is essential to provide a managed distributed data access with a central metadata server and full support for VO-based access control and authorization. While we have not yet implemented such a solution, it is an essential element to allow international collaboration. We have been able to test a number of existing solutions for grid infrastructure and are confident that a metadata service on the base of AMGA, the Arda Metadata Catalogue Project (cf. Santos [19], Piparo [18]), would be a viable solution that could allow us to leverage rich metadata services and grid access controls to enable linguistic researches to use the available resources while enforcing the legal restrictions in place.

VO Execution environments. For testing purposes, we have developed a set of command-line tools for typical linguistic grid jobs and prepared execution environments with all the necessary software packages pre-installed. These tools already provide a way to perform resource-intensive tasks using distributed corpus data and distributed computing resources in the HLT VO. We are planning to expand this tool set and to develop it into a viable basis for the future use in the new VO and to use it to develop more advanced tools. Furthermore, we are planning to build a set of web services and web grid interfaces to enable linguists to use the new tool-set with ease. The final form of the HLT VO execution environment is not yet decided as we hope it will be shaped according to the needs and requirements of future member organizations.

Web interfaces and central services. A dedicated web site for information, documentation and user management of HLT VO is being set up at JSI as part of Slovenian National Grid Initiative effort. It will provide the central grid services for the VO, such as basic task and job reporting, statistics of usage and meta-data access. The central infrastructure will soon be sufficient for initial testing and evaluation for Human Language Technologies Grid, but additional services will have to be developed to support web based job submission and control, data-set upload (including corpus upload, transformation etc.) and data retrieval from finished jobs. A number of these techniques have been tried in the experiments we described previously.

8 Conclusions

We have tried to demonstrate with our experiments that the use of grid infrastructure as the basis of a new, more powerful and flexible tool-set, is viable and, in fact, brings many advantages over the traditional methods.

One of the major attractions of the new system, next to the flexibility, compatibility of tools and the sheer computing and storage power, will be to provide a single method (and programming API) to many resources in different languages, and to resolve the difficulties inherent in different legal, technical and practical restrictions that make any multilingual research rather difficult today.

We hope that many new communities will join us and contribute to make the toolset and interfaces of HLT VO as flexible and as powerful as possible, and, by contributing their national corpora and other resources, will enable a new, rich multilingual linguistic research community to develop new tools and new approaches and to advance in development of existing approaches due to new possibilities for comparison of tools and methods across language barriers. In this manner, we hope to not only overcome linguistic and administrative barriers, but to build from the multitude of solutions and approaches a more powerful and flexible toolset for future research.

References

- [1] Banerjee, S., Pedersen, T. (2003). The Design, Implementation, and Use of the Ngram Statistics Package. In Proc. of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, February 17–21, 2003, Mexico City
- [2] Brants, T. (2002). TnT—A Statistical Part-of-Speech Tagger. In Proceedings of the 6th Applied Natural Language Processing Conference ANLP-2000 (pp. 224–231). Seattle, WA.
- [3] Carroll, J., Evans, R., Klein, E. (2005). Supporting text mining for e-science: the challenges for grid-enabled natural language processing. In: Proceedings of the UK e-Science All Hands Meeting.
- [4] Ellert, M., et al. (2007). Advanced Resource Connector middleware for lightweight computational Grids. *Future Generation Computer Systems* 23, pp. 219–240.
- [5] Erjavec, T. (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. Fourth International Conference on Language Resources and Evaluation, LREC'04. (pp. 1535-1538). ELRA, Paris.
- [6] Erjavec, T. (2007). An Architecture for Editing Complex Digital Documents. In Proc. of the 1st Intl. Conference “Digital information and heritage”. Zagreb, 2007, pp. 105–114.
- [7] Erjavec, T. and Džeroski, S. (2004). Machine Learning of Language Structure: Lemmatising Unknown Slovene Words. *Applied Artificial Intelligence*, 18/1 (pp. 17–41). Taylor & Francis.
- [8] Erjavec, T., Ignat, C., Pouliquen, B., Steinberger, R. (2005). Massive multi-lingual corpus compilation: Acquis Communautaire and totale. *Arch. Control Sci.*, vol. 15, pp. 529–540.
- [9] Erjavec T, Javoršek, J. J. (2008). Grid Infrastructure Requirements for Supporting Research Activities in Digital Lexicography. *Mondilex: Lexicographic Tools and Techniques*. Moscow, IITP RAS, pp 5–13.
- [10] Erjavec, T., Krek, S. (2008). The JOS morphosyntactically tagged corpus of Slovene. In Proc. of the 6th International Conference on Language Resources and Evaluation, Marrakech, Morocco, May 26 - June 1, 2008. LREC 2008. Marrakech: ELRA
- [11] Luís, T., Martins de Matos, D., Paulo, S., Daniel Ribeiro, R. (2008). Natural Language Engineering on a Computational Grid (NLE-GRID) T5 - Performance Experiments, Tech. Rep. 35 / 2008 INESC-ID Lisboa, January 2008.
- [12] Manandhar S., Džeroski S. and Erjavec T. (1998). Learning Multilingual Morphology with CLOG. In Proceedings of Inductive Logic Programming; 8th International Workshop ILP-98 (Lecture Notes in Artificial Intelligence 1446) (pp. 135–144). Springer-Verlag, Berlin.
- [13] Martins de Matos, D., Tiago Luís, Daniel Ribeiro, R. (2008a). Natural Language Engineering on a Computational Grid (NLE-GRID) T1 - Architectural Model, Tech. Rep. 30 / 2008 INESC-ID Lisboa, January 2008

- [14] Martins de Matos, D., Daniel Ribeiro, R., Paulo, S., Batista, F. Coheur, L., Paulo Pardal, J. (2008b). Natural Language Engineering on a Computational Grid (NLE-GRID) T2 - Encapsulation of Reusable Components, Tech. Rep. 31 / 2008 INESC-ID Lisboa, January 2008.
- [15] Martins de Matos, D., Daniel Ribeiro, R. (2008c). Natural Language Engineering on a Computational Grid (NLE-GRID) T2h - Encapsulation of Reusable Components: Lexicon Repository and Server, Tech. Rep. 32 / 2008 INESC-ID Lisboa, January 2008.
- [16] Marujo, L. Lin, W. Martins de Matos, D. (2008). Natural Language Engineering on a Computational Grid (NLE-GRID) T3 - Multi-Component Application Builder, Tech. Rep. 33 / 2008 INESC-ID Lisboa, January 2008.
- [17] Neuroth, H., Kerzel, M., Gentzsch, W. (eds.), (2007). German Grid Initiative D-Grid.
- [18] Piparo, D. (2007). Two graphical browsers for the AMGA metadata catalogues. In Nuclear Physics B - Proceedings Supplements, vol. 172, pp. 311–313.
- [19] Santos, N. and Koblitz, B. (2008). Security in distributed metadata catalogues. *Concurrency and Computation: Practice and Experience* 20 (17), 1995-2007.
- [20] Sperberg-McQueen, C. M. and Burnard, L. (eds.) (2002). Guidelines for Electronic Text Encoding and Interchange, the XML Version of the TEI Guidelines. The TEI Consortium.
- [21] Tamburini, F. (2004). Building distributed language resources by grid computing. In Proc. of the 4th International Language Resources and Evaluation Conference. pp. 1217–1220.
- [22] Vintar, Š. (2001). Using Parallel Corpora for Translation-Oriented Term Extraction. *Babel* 47(2), John Benjamins Publishing.

Slovak paremiography database*

Peter Ďurčo¹ and Radovan Garabík²

¹ Univerzita sv. Cyrila a Metoda v Trnave, Trnava

² L. Štúr Institute of Linguistics, Slovak Academy of Sciences, Bratislava

Abstract. The article describes the effort to design a wiki-based paremiography database and the process used to extract data from a published dictionary of proverbs. The database is build using MoinMoin engine, offering (standard) possibilities of full text search, categorisation, on-line editing and access control lists. The process used in parsing and correcting the OCRed source is described in detail, and most common sources of errors are discussed.

1 Introduction

A paremiography dictionary (or a database) spreads our lexicographic description of a language into a broader realm of commonly used expressions, and as such, it extends and complements the (better researched and described) dictionaries of idioms.

Such a dictionary is paralleled by a collocation dictionary[2] – proverbs can be seen as a subset of collocations, however, while a proverb is a self-contained, independently functioning language unit, a collocation can be nothing more than almost a random, high frequency sequence of words.

Concerning Slovak language, so far unsurpassed paremiography collection is a compilation by Adolf P. Zátarecký [6], first published in 1896. It contains over 10 000 different proverbs (not counting variants). The influence of this work on any subsequent paremiography compilations was immense, since no other collection came even close to the volume of this work, and there was virtually no need to engage in additional field research – following compilations just upgraded and refined selected subsets of Zátarecký’s collection. The collection itself has been reprinted several times (with the orthography and language progressively converted to ever increasingly modern Slovak, acquiring additional notes and comments), the most recent edition was published as late as in 2006 [7].

The core of the collection is made up of proverbs, sayings and locutions. However, there are also some more indefinite units (pieces of weather-lore, rhymes etc.) as well as other types of phraseologisms (similes, figurative expressions). Although the collection does not record phraseology in its entire extent but concentrates on one type of idioms – proverbs and sayings, i. e. stable sentences. Zátarecký divided the entire material into 20 thematic groups (man, one’s age, sex, family and home, human body, its needs, disease and death, social circumstances, social classes, status, descent and employment, possession and nourishment, food, clothes, cleanliness and dance, human intellect, general rules of wisdom and carefulness etc.). The collection includes immensely valuable material which is however only insufficiently exploited and explored from the point of view of linguistic theory and interdisciplinary research. Zátarecký tried to solve the problem of variability of proverbs. His correspondence with other scholars gives also evidence of his interest in the semantics and etymology of proverbs. Zátarecký, together with Dobšínský dealt also with paremiological terminology and they attempted to elaborate optimal taxonomy of thematic concepts. Zátarecký combined an alphabetical order of statements within the thematic groups. He also applied the formal criterion of division within particular groups and elaborated the index of key words.

The goal of our effort is to put existing proverbs (not just from Zátarecký’s collection) into an easily searchable electronic database, to aid further research and to have a ready source of information. Since the Zátarecký’s collection was available only in printed form, this article deals

* The study and preparation of these results have been partly supported by the EC’s Seventh Framework Programme [FP7/2007-2013] under the grant agreement 211938 MONDILEX.

mostly with specific issues connected with converting its scanned version into a machine readable form, while keeping most of the available information.

2 Implementation

The database has been implemented as a straight, unmodified MoinMoin installation³. Since the database is expected to be pre-filled with the data, it will be used mostly in passive mode (searching the data) and the editing will be limited to occasional fixing of typos and OCR errors, we do not need to concern ourselves with designing an additional user-friendly data visualization and/or editing. The database micro- and macrostructure is implemented only in a set of guidelines for the users, concerning article structure and components, while keeping standard MoinMoin syntax (in fact, only a tiny subset of it, to facilitate further automatized article parsing).

The wiki engine is centered on the concept of ‘pages’ – each page keeps a separate, contained information, is uniquely identified by its name and can optionally belong to one or more categories. Our database maps one (semantic) locution into one wiki page. The page starts with locution variants, separated by an empty lines (visualised as separate paragraphs), followed by an optional comment (currently used to note the locution number in Záturecký’s collection, if applicable), followed by a list of categories the locution belongs to (see Tab. 1).

```
<entry> ::= <locution> {<p> <locution>} \n ---- \n [ <comment> ]
          \n ---- \n <category> { <category> }
<locution> ::= ? characters ?
<comment> ::= ? characters ?
<p> ::= \n \n {\n}
<category> ::= Category ? characters ?
```

Table 1. Formal description of an entry syntax

Initially, the core of the database consisted of proverbs from [4], extended by selected proverbs from [3, 5], giving first 2828 entries, then we added Chapter 3 of Záturecký’s collection.

3 Deriving a page name

In the first version of the database, we kept the name of the wiki page to be the complete locution, including proper capitalization and punctuation. This has one undisputed advantage – when using the built-in MoinMoin search, searching for a given word will return all the pages that contain the word in their title (Fig. 1).

However, we soon hit several problems:

- There are often several variants of a given proverb. It is desirable to keep all the variants clumped together, and by listing the variants inside one page we would loose the search ability. The situation however could be remedied by choosing one of the variants as the ‘main’ one and keep the others as redirect pages.
- There is a technical limit for maximum page name length, arising from underlying filesystem limitation – MoinMoin stores the pages as files, with file names being an ASCII quoted version of page name (each unsafe character is stored as its UTF-8 representation in hexadecimal, enclosed in parenthesis). No matter what the actual filesystem limit, traditional Unix file

³ <http://moinmo.in>

system (used in BSD variants) and the Linux VFS place a hard limit of 255 bytes for the file name length – many proverbs are simply longer than that.⁴

- We need to put the proverb inside the page content as well – repeating the same information in the page name and page content is redundant and prone to errors, and makes automatized manipulation with the data cumbersome (we have to watch two instances of the same information).

In the second iteration of the database, we decided to use different, unique page names. Out of several choices available (numbers, random strings, various transformations of locutions) we decided to choose a ‘semantic hash’, trying to reduce the locution down to as little words as possible, while keeping a hint of the meaning in the resulting name.

At a first glance, the most obvious thing to do with the locution is to lemmatise its constituent words, to get their basic forms, without ballast of additional grammar information. There are, however, two main problems arising from this approach. First, Slovak exhibits relatively high level of homonymy, so the texts should undergo also morphology disambiguation. That is, unfortunately, inherently imperfect process, with accuracy rarely exceeding 95 %. The problem is exasperated by the fact that morphology disambiguation is usually tuned to ‘normal’ texts (and therefore is even less precise for our locutions, with their specific kind of language) and the embarrassing nature of a bad lemmatisation – which would completely ruin the meaning of the page name⁵. The second problem is that proverbs are ‘semi-frozen’ expressions, with the grammar categories often strictly given for a given locution, and lemmatising would blur the traditional wordforms used and diminish the usefulness of semantic hints the page name could provide. At the end, we decided not to try to lemmatise the page names.

The second most obvious process is to eliminate ‘unimportant’ words. We keep not only lexical words (such as nouns, verbs, adverbs, adjectives), but also prepositions and two words *sa* and *si*. The (somewhat surprising) presence of preposition is necessitated by not lemmatising the nouns – the case is often governed by prepositions and excluding the preposition would lead to markedly ungrammatical sentences. *Sa* and *si* form (among other possibilities) a part of reflexive verbs, and leaving out an obligatory reflexive marker would again emphasise ungrammaticality.

To keep the page names short, we include at most two words that are either noun or verb (with the exception of forms of verbs *mat*, *byť* and *jest*⁶). If there is a locution to be added to the database and the derived page name already exists, we keep adding another (lexical) words until the page name is unambiguous. This algorithm has an advantageous side effect: it quite reliably detects duplicates that differ mostly in function words.

4 Structure of the original text

Zátarecký’s collection is divided into 20 chapters (19 in the edition [7]), each concerning certain aspect of society or language. Locutions in each chapter are numbered, starting with number 1, with every 5th entry marked at the left text margin. Different typeface (and a smaller font size) is used for literature and references – this style is recognised by the OCR software as italics. Locution variants are (somewhat unfortunately) separated by a dash surrounded by spaces: ‘ – ’.

Each chapter is accompanied by a comments section, containing further explanation of the locutions, often including Hungarian, German, Polish or Latin equivalents, or explanation of dated terms and expressions, otherwise incomprehensible for a contemporary reader (these comments were often not written by Zátarecký himself, but by later editors). The comments are numbered by the number of the locution they refer to.

⁴ Note that the new storage backed system being prepared for MoinMoin v. 2.0 is going to lift these limitations, since the pages will no longer be necessarily stored as corresponding single files in the filesystem.

⁵ Given especially the two meaning of the word *mat*, ‘mother’ and ‘to have’, words that thanks to their nature occur very frequently in proverbs.

⁶ ‘to eat’, 3rd person singular *je* is homonymous with the same categories of the verb *byť*, ‘to be’

Výsledky 1 - 10 z 11 výsledkov z približne 3080 stránok. (0.61 sekúnd)	
1. Krátka reč i pekné slovo vymôže u pánov mnoho.	0.1k - revízia: 1 (aktuálny) posledná zmena: 0
2. Neťahaj si slovo naspät!	0.1k - revízia: 1 (aktuálny) posledná zmena: 0
3. Pekným slovom kedy-tedy i psa utišiš.	0.1k - revízia: 1 (aktuálny) posledná zmena: 0
4. Rana sa zahojí, ale slovo nie.	0.1k - revízia: 1 (aktuálny) posledná zmena: 0
5. Skôr od jalovej kravy tela vydrapí, ako od toho slovo.	0.1k - revízia: 1 (aktuálny) posledná zmena: 0
6. Slovo je viac ako závdavok.	0.1k - revízia: 1 (aktuálny) posledná zmena: 0
7. Slovo robí muža.	0.1k - revízia: 1 (aktuálny) posledná zmena: 0
8. Zlé slovo iba tomu za väzy padá, kto klaje.	0.1k - revízia: 1 (aktuálny) posledná zmena: 0
9. Zo sprostej hlavy sprosté slovo.	0.1k - revízia: 1 (aktuálny) posledná zmena: 0
10. Človeka chytajú za slovo, vola za rohy.	0.1k - revízia: 1 (aktuálny) posledná zmena: 0

1 2 Ďalší

Fig. 1. Searching for a word 'slovo' in all the page titles, in the old version of the database. Later, we have abandoned the idea of using complete locutions as page names.

Výsledky 1 - 10 z 14 výsledkov z približne 4291 stránok. (5.62 sekúnd)	
zo sprostej hlavy sprosté slovo	... 2 zhody
...Zo sprostej hlavy sprosté slovo. ---- [[CategoryCore]]...	
0.1k - revízia: 1 (aktuálny) posledná zmena: 0	
zlé slovo	... 2 zhody
...Zlé slovo iba tomu za väzy padá, kto klaje. ---- [[CategoryCore]]...	
0.1k - revízia: 1 (aktuálny) posledná zmena: 0	
slovo závdavok	... 2 zhody
...Slovo je viac ako závdavok. ---- [[CategoryCore]]...	
0.0k - revízia: 1 (aktuálny) posledná zmena: 0	
slovo robí	... 2 zhody
...Slovo robí muža. ---- [[CategoryCore]]...	
0.0k - revízia: 1 (aktuálny) posledná zmena: 0	
pekným slovom i psa	... 2 zhody
...Pekným slovom kedy-tedy i psa utišiš. ---- [[CategoryCore]]...	
0.1k - revízia: 1 (aktuálny) posledná zmena: 0	
neťahaj si slovo	... 2 zhody
...Neťahaj si slovo naspät! ---- [[CategoryCore]]...	
0.0k - revízia: 1 (aktuálny) posledná zmena: 0	
neber slovo	... 2 zhody
...Neber každé slovo na vážku. Záturecký 161 ---- [[CategoryZátureckýPomerySpoločenské]]...	
0.1k - revízia: 1 (aktuálny) posledná zmena: 0	
krátka reč i pekné slovo	... 2 zhody
...Krátka reč i pekné slovo vymôže u pánov mnoho. ---- [[CategoryCore]]...	
0.1k - revízia: 1 (aktuálny) posledná zmena: 0	
žena musí	... 1 zhoda
...Žena musí mať posledné slovo. ---- [[CategoryCore]]...	
0.1k - revízia: 1 (aktuálny) posledná zmena: 0	
človeka chytajú	... 1 zhoda
...Človeka chytajú za slovo, vola za rohy. ---- [[CategoryCore]]...	
0.1k - revízia: 1 (aktuálny) posledná zmena: 0	

1 2 Ďalší

Fig. 2. Fulltext search for a word 'slovo'.

5 Scanning and conversion

Záturecký’s collection has been scanned and submitted to OCR using the ABBYY Finereader software⁷. As a result we obtained document in Microsoft Word (97–2003) format, including distinct typographic styles, as present in the original scanned text. As the text style differences encode useful information, it is desirable to keep this information (it can also aid in correct parsing), therefore we needed to convert the document into some kind of a structured, easily readable format. An example of such a format is the ubiquitous XML, and since the OpenOffice uses natively OpenDocument format (XML based) [1], an obvious approach would be to open the Microsoft Word file in the OpenOffice Writer and save it as the OpenDocument text, which can be further parsed using standard XML parsing tools and libraries. Unfortunately, the conversion does not deal well with character styles – the resulting document contains 235 common styles, majority of them used to encode the same basic paragraph style. In order to parse the document, we would have to infer the original text style for each of the OpenOffice character styles, which would be a rather time consuming process.

A better approach was to convert the file into HTML, which marks the text styles into either different FONT tags, distinguished by the `size` attribute, or B and I tags for boldface and italics, respectively. It is worth pointing out, that due to OCR errors and imperfections, the styles are not always recognised correctly, and it has to be taken into account during parsing. To parse the HTML file, we have chosen the BeautifulSoup framework⁸, which is a Python HTML/XML parser with easy API and emphasis on parsing damaged or invalid HTML/XML files. Although the HTML file we obtained is perfectly valid, in the course of processing we split the file at some specific points, sometimes obtaining HTML chunks with unpaired tags, where we conveniently use BeautifulSoup’s abilities to deal with invalid HTML.

Since it is highly desirable to link comments to the proper locutions, we needed to keep correct numbering of the entries. We start by splitting the HTML file at positively identified locution index numbers – numerals dividable by 5, in italics and using smaller font size. We take into account only monotonically increasing numbers, since we have to realise that due to OCR errors, there are going to be gaps and other errors in the sequence (OCR is good in finding out sequences of numeric characters, therefore errors caused by replacing the digit ‘1’ by lowercase letter ‘l’ occur only with standalone characters, rarely in multi-digit numerals, however errors produced by replacing ‘1’ with ‘7’, ‘8’ with ‘6’ or ‘9’ and vice versa are common), and we disregard numbers that differ from previous index by more than 40 (since that is probably an OCR error).

Once we have these HTML chunks with the range of entry numbers for each of them, we split them by additional present entry index numbers (if their span is more than 5 indices), this time regardless of their visual style (which is often incorrect). With these smaller chunks, we already used up all the information that there was about the entry numbers, and we had to apply some heuristics to split the remaining locutions correctly. We conveniently used the ability of BeautifulSoup to parse invalid HTML and fed the chunks back into the parser. This time we looped through the P tags and separated texts from the paragraphs (which correspond to locutions). Since the OCR does not find ends of paragraphs very reliably, in the next step we joined back the sentences where the paragraph started with a lower case letter. Then we tried to find out sentence boundaries and split the text on them (fixing the cases where the OCR ignored paragraph break). The heuristics is simple – a sentence boundary is where the previous character is one of full stop, exclamation mark or question mark, and the next character is an uppercase letter. This fails in some cases (e.g. single expression *Koho bili? Petra. A kto sa bil? Peter.* will be incorrectly separated into 4 locutions), but overall significantly improves the segmentation. Then a plain text file with numbered locutions had been produced. We then apply Procrustean bed method to keep the locutions properly numbered: if the number of automatically segmented locutions is smaller than expected according to index numbers, we put dummy sentences consisting of a single character (we have chosen ‘@’) into the file; if the number is bigger, we join excessive ones, again using character ‘@’ as a separator.

⁷ <http://www.abbyy.ru/finereader/>

⁸ <http://www.crummy.com/software/BeautifulSoup/>

After the segmentation, entry numbers were manually corrected. The process consisted basically of finding all the occurrences of the ‘@’ character and splitting or joining the lines as required. In Chapter 3, out of 1405 locutions, there were 227 incorrectly numbered ones (each one of the incorrectly parsed locutions is counted twice, first at its original number, second time the for the locution number it replaced). One of them was caused by genuinely dropped number in the printed source, 168 occurrences were caused by transposed pages in the OCRed text – that gives only 4.2% genuine error rate, with a very quick manual correction (we have to stress here that we were not proofreading the text and fixing OCR typos, just fixing the numbering of entries).

Fixed text is then parsed again and converted into internal MoinMoin structure. For the comments, we conveniently used the subpages MoinMoin mechanism – if present for a given locution, each comment is been put into a subpage named /poznámka⁹, with a link from the parent (locution) page. Since many of the comments were written by subsequent editors of Záturecký’s collection, their copyright protection has not expired yet, and we cannot make them freely available. We used the MoinMoin’s possibility to use ACL to block public access to these subpages. In the Chapter 3, there were 253 comments present.

The locutions are categorised – there is a category for the Chapter 3 locutions¹⁰, a category for the ‘core’ proverbs¹¹ and a category for comments to Záturecký’s locutions¹². Given page can belong to more than one category, as a matter of fact, many of the core proverbs also belong to `CategoryZátureckýPomerySpoločenské`.

6 Conclusion

Presented database is intended to serve as an easily reachable source of paremiography data. To test the concept, Chapter 3 of Záturecký’s collection [6, 7] has been scanned, OCRed and converted to the database, together with a few thousand other selected proverbs. The conversion process is mostly automatic, with minimal (though still substantial) human intervention, and will be used to convert remaining chapters of the collection.

⁹ i.e. comment

¹⁰ `CategoryZátureckýPomerySpoločenské`

¹¹ `CategoryCore`

¹² `CategoryZátureckéhoPoznámky`

References

- [1] ISO/IEC 26300:2006 (2006). *Information technology – Open Document Format for Office Applications (OpenDocument) v1.0*. Geneva: International Organization for Standardization.
- [2] Majchráková, D. & Ďurčo, P. (2009). Compiling the First Electronic Dictionary of Slovak Collocations. To be published.
- [3] Miko, F. et al. (1989). *Frazeológia v škole*. Bratislava: Slovenské pedagogické nakladateľstvo.
- [4] Mlacek, J. & Profantová, Z. (1996). *Slovenské príslovia a porekadlá, zv. 1–2. Výber zo zbierky A. P. Zátureckého*. Bratislava: Nestor.
- [5] Smiešková, E. (1988). *Malý frazeologický slovník*. Bratislava: Slovenské pedagogické nakladateľstvo.
- [6] Záturecký, A. P. (1896). *Slovenská príslovia, porekadla a úsloví*. Praha: Česká akademie věd.
- [7] Záturecký, A. P. (2006). *Slovenské príslovia, porekadlá, úsloví a hádanky*. Bratislava: Slovenský Tatran.

The Japanese-Slovene Dictionary jaSlo: a usability study

Kristina Hmeljak Sangawa¹ and Tomaž Erjavec²

¹ University of Ljubljana

² Jožef Stefan Institute

Abstract. The paper presents the on-line Japanese-Slovene dictionary jaSlo, in particular the ways in which it has been used, and how it has been extended with examples mined from a parallel corpus. The paper first describes jaSlo and the structure of its dictionary entry, its Web interface for searching, and an analysis of the access logs. The use of jaSlo in the context of the Japanese reading-support tool Reading Tutor is described next, again followed by an analysis of the access logs. Also discussed is the manner in which usage examples were added to the dictionary, and an evaluation of their usefulness. The paper concludes with directions for further work.

1 Introduction

The establishment of the first program of Japanese studies in Slovenia at the University of Ljubljana in 1995 brought with it the need for Japanese language teaching materials and dictionaries for Slovene speaking students. However, due to the very small number of potential users, probably not much more than the cca. 300 students and graduates of this Japanese studies course, the production of a Japanese-Slovene dictionary is not a particularly profitable project that could interest a publishing house. The teachers of the course therefore decided to create a dictionary and progressively publish it on the web while it is being enhanced, continuously striving to fully exploit available resources for this low-budget project. The first version was published in 2002 [1], in 2003 it was converted into a TEI compliant XML format and moved to a server at the Institute Jožef Stefan [2]. The 3rd version released in 2006 added more information to the dictionary entries, mostly acquired via third party resources [3], and added more entries: it has approximately 10.000 Japanese lemmas with about 25.000 Slovene translational equivalents. The dictionary is available for searching at the address <http://nl.ijs.si/jaslo/>.

Simultaneously with enlarging the dictionary we also enhanced its content, by inclusion of publicly available data and programs, automatic collection of examples from a bilingual parallel corpus and from a web-harvested corpus. We also included the dictionary into the Japanese reading-support tool Reading Tutor [4].

Both tools, Reading tutor and the Web interface to the jaSlo dictionary, keep a log of user lookups. Reading Tutor's log records include the date and time of access and the text looked up, while the dictionary jaSlo records the date and time of access, the string looked up, and the number of returned hits.

In this paper we present in Section 2 the jaSlo dictionary, its compilation, structure and means of access, and then give an analysis of user logs, in order to discover what aspects of the tool need improvement. We then move in Section 3 to the Reading Tutor application, give a brief introduction and, again, an analysis of user logs. Section 4 deals with current work on enhancing the dictionary with usage examples automatically extracted from a Japanese-Slovene parallel corpus. We explain the corpus compilation process and how the example are included into a (test version) of the dictionary. Section 5 concludes with some directions for further work.

2 The jaSlo Dictionary

The jaSlo dictionary began as a set of separate small glossaries prepared by the professors and students at the Department of Asian and African Studies of Ljubljana University, which were mostly in tabular and HTML format. These glossaries were first converted into a common encoding, the dictionary entries from the separate files merged, and manually checked. The target encoding took into account international standards in the field, which brings with it a number of well-known advantages, such as better documentation, the ability to validate the structure of the document, simpler processing, easier integration into software platforms, longevity and easier Web deployment. It is available on the Web, via a search interface which keeps a log of user accesses.

2.1 Dictionary structure

For encoding the dictionary we use the XML version of the Text Encoding Initiative Guidelines, TEI P4 [5] in particular its module for dictionary encoding.

```
<entry id="jaslo.6557">
  <form type="hw">
    <orth type="kana">ちようせつする</orth>
    <orth type="kanji">調節する</orth>
    <orth type="roma">chousetsusuru</orth>
  </form>
  <gramGrp><pos>Vs</pos> <subc>trans.</subc></gramGrp>
  <trans><tr>uskladiti</tr> <tr>uravnati</tr> <tr>uravnavati</tr></trans>
  <eg>
    <q>室内 ( しつない ) の温度 ( おんど ) をちようせつする</q>
    <tr>uravnavati temperaturo v sobi</tr>
  </eg>
  <xr type="lesson" n="L1.23"><xref>1. letnik, lekcija 23</xref></xr>
  <usg type="level">0</usg>
  <note type="admin" resp="TER">2005-07-11 Add romaji</note>
  <note type="admin" resp="TER">2005-07-10 Add levels</note>
  <note type="admin" resp="ISE">2005-02-28 Merge</note>
  <note type="admin" resp="VOJ">2005-02-22 V (440)</note>
  <note type="admin" resp="KHS">2003-03-12 L1 (850)</note>
</entry>
```

Figure 1. A typical dictionary entry in jaSlo

Figure 1 presents a typical dictionary entry in jaSlo, which first includes the `<form>` for the headword. The form is given in three scripts: kanji, kana (hiragana or katakana) and in transcription to Latin, so called romaji. This is followed by grammatical information, translation into Slovene, examples, a reference to the lesson where the word is introduced, the difficulty level of the entry according to the Japanese Language Proficiency Test Specifications [6], and finally administrative information tracing the compilation history. In addition to the elements given in the example, the following information is also present in a subset of the entries: cross-reference to related entries (esp. synonyms with different levels of politeness etc.), inflected forms of verbs, the etymology of loan-words, and encyclopaedic descriptions of proper names and Japanese culturally bound terms.

It should be noted that quite some time was spent devising the grammatical tagset (content of `<gramGrp>`), and then semi-automatically converting the legacy labels to this common standard. Our set of categories contains 19 different labels and is based on the set used in the Japanese morphological analyser Chasen [7], which makes it easier to use jaSlo with Chasen tagged corpora.

2.2 Using the dictionary

The dictionary is deployed via a Web-based interface, available at <http://nl.ijs.si/jaslo/>, which allows full text searches by string or word on the dictionary, with optional restriction of the match to headword or translation, and filtering by PoS or difficulty level. The interface is also localised to Slovene, Japanese and English. The user's browser is assumed to offer Unicode support and have a Japanese-language font installed but, apart from that, no requirements are imposed on the client architecture. The server is implemented as a Perl CGI script, which accepts the search parameters and sequentially, via a SAX filter, returns the entries that match the query, using an XSLT stylesheet similar to the one used by the editors, but which ignores certain information, e.g. admin notes, entry ID etc. While this means that for each query the complete dictionary has to be processed, this does not present problems with the current size of the dictionary and user load.

The dictionary can be searched from the interface with the following options (c.f. Figure 2): whole text

search, search by word class (nouns, verbs, adjectives), headwords only. While the dictionary was conceived as a Japanese-Slovene dictionary and its Slovene-Japanese counterpart is planned for a later stage, it can already be used to look up translations of Slovene words by entering a Slovene search string and searching through the entire dictionary text (headword translations and examples), although the information thus obtained can be confusing when there are numerous Japanese headwords with the same Slovene translation.

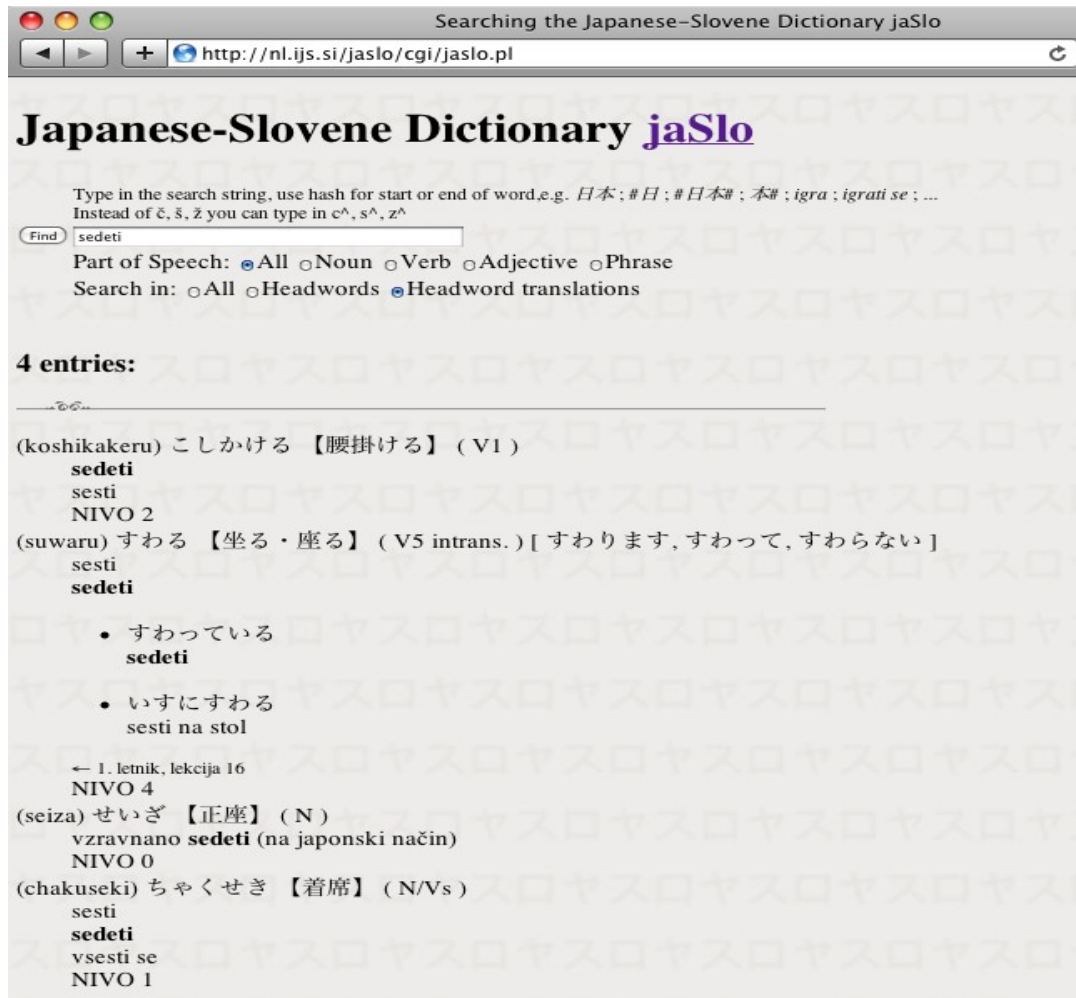


Figure 2. Search interface and display of results

2.3 Analysis of lookups in jaSlo

Each query to jaSlo is logged together with the time and number of returned entries (without client machine address, thus preserving privacy), which enabled us to begin tailoring the dictionary to user needs. This section presents an analysis of the log file in order to guide our further work on the dictionary.

From September 2006 (when the current 3rd version of the dictionary was released) to April 2008, 19,579 lookups were recorded, corresponding to 11,938 search string types. Most strings were single words, but quite some phrases were also found as input strings (e.g. “všeč si mi” (I like you), “ime mi je” (my name is), “vse najboljše za rojstni dan” (happy birthday), “ikaga desu ka” etc.). Here a more clear explanation of the function of each of the two tools on this server (Reading tutor for longer texts, jaSlo for single words or multi-word units), would help users avoid this kind of unhelpful searches.

Less than half of the search strings were in Japanese characters, as can be seen in Table 1.

hiragana	33%
kanji or kanji-kana mixture	20%
katakana	5%
romanized Japanese words	17%
Slovene words	24%
proper names	0.6%
English words	0.4%

Table 1. Percentage of search strings by character type

Possible explanations for this massive use of romanized Japanese are that users are not comfortable with typing Japanese (beginners or students with little typing experience) or that they access the dictionary from terminals with no Japanese script support (on public terminals in libraries, internet cafés etc.). Our dictionary includes romanized forms of all headwords alongside their kana and kanji forms, and the very large amount of romanized Japanese search strings confirmed our choice of including them. However, since only headwords contain Latin script forms, it might be useful to romanize the whole Japanese content of the dictionary (inflected forms and examples) in order to improve the search hit ratio.

Given the considerable number of searches for proper names (mostly Slovene personal names), it might be useful to include katakana forms of the most common Slovene names, as well as Japanese proper names which could be easily obtained from freely available vocabulary lists (e.g. Unidic). Searches for English words only return a hit when they appear in an etymological note for katakana words, since the dictionary does not include English otherwise, so users must have soon realised that this is not an English dictionary.

There were even two Hangul and two Arabic search strings, and a few meaningless character strings, but overall users seem to be using the dictionary for what it is made for: looking up Japanese words, often also Slovene words. The top 20 search strings are given in Table 2.

search string	No. of searches	translation	search string	No. of searches	translation
dober dan	82	good day	medved	32	bear
ljubezen	67	love	hiša	32	house
dan	44	day	a	26	
ljubim te	41	I love you	pes	25	dog
日本	36	Japan	igra	24	game
zdravo	36	hallo	tsuku	23	reach, be attached
hvala	35	thank you	mesto	21	city, place
jaz	34	I	san	20	Mr./Ms.
avto	34	car	love	20	
sonce	32	sun	ljubiti	20	to love

Table 2. Most searched-for strings

The very small number of Japanese search strings among the top 20 is mostly due to the fact that each Japanese word can be and presumably was searched for in different forms: latin script, hiragana or kanji, which are counted as separate searches.

We were particularly interested in search strings which did not return any hits, because these are ideal candidates for inclusion in our next dictionary revision. Many words were found which were not included

in the dictionary because they do not figure in Japanese vocabulary frequency lists, although they denote concepts which are used rather frequently in Slovenia (čebela - bee, šipek – rosehip, stalagmit – stalagmite etc.), or simply interesting to the users of our dictionary (prevajalec – translator, pozitivna energija – positive energy, idiot – idiot etc.).

Many zero hit logs were searches of Japanese words in latin script while the function “search Slovene translations only” was on, or searches of words in the wrong word class, e.g. searching the word カボチャ (cabocha “pumpkin”) among verbs etc. Such searches returned 0 results although words were actually in. Here a simpler default user interface (combined with a non-default advanced interface where searches could be limited to determined categories, as in our first interface), a function which looks up words in fields other than headwords, or a fuzzy search function when no result is found in a limited part of the dictionary might help avoid such problems.

A third cause for missed hits was the use of capitals: the search mechanism of our dictionary is cap-sensitive, so that e.g. “IGRA” in capital letters returns 0 hits, while “igra” in small letters returns a few appropriate lines. Given the rather erratic use of capitals in search strings, it would be better to make searches cap-insensitive.

This analysis of user logs thus confirmed some of our choices, foremost the decision to log all queries and keep a track of the way the dictionary is being used, but also pointed out possible improvements which could make the dictionary more user-friendly: a clearer explanation of the function of each tool on the same server, a simpler user interface, less restrictive default search options, more comprehensive romanisation and the inclusion of words which do not appear in general Japanese vocabulary lists.

3 Reading Tutor

The "Reading Tutor" (<http://language.tiu.ac.jp/>) is a Web based on-line Japanese reading support system composed of a dictionary tool, a level detection tool, and a collection of learning materials and quizzes. The dictionary tool analyses any text input by on-line users using the Japanese morphological analyzer Chasen [7], links every token in the text to one of Reading Tutor's dictionaries (Japanese definitions, Japanese-English and Japanese-German in the original version), and then presents the hyperlinked text alongside a glossary of all words it contains. Users can then read through the text and summon up readings and meanings of unknown words by simply clicking on them.

The Reading Tutor lexica are encoded in XML, according to their own document type. The Reading Tutor DTD is quite complex, with numerous elements, quite a few of them required. In order to include jaSlo into Reading Tutor we wrote an XSLT stylesheet that converts our TEI encoding into the schema for Reading Tutor; the jaSlo dictionary was then added to the Reading Tutor, and in 2007 a mirror server was set-up at <http://nl.ijs.si/jaslo/chuta/> and an screenshot example is given in Figure 3.

2.4 Analysis of lookups in Reading Tutor

As with jaSlo, we log the accesses to Reading Tutor, by recording the time of the request and the submitted text. In its first year of public access, from May 2007 to April 2008, Reading Tutor's Slovene module has recorded 592 lookups, scattered over 153 days in the whole year, with an average of slightly less than 2 accesses per day and a peak of 44 accesses on 2nd August 2007. Considering that in Slovenia there are presumably not more than 400 Slovenian speaking learners of Japanese at present, these rather modest figures are not very surprising, but they do indicate that the service needs some more publicity.

Especially in the first months of operation there were many access logs of texts which were clearly input only in order to try how the tool works: chunks of words copied from Reading Tutor's homepage itself, very basic words like **ありがとう** (arigatou “thank you”), **こんにちは** (konnichiwa “hello”), **日本語** (nihongo “Japanese language”) etc.



Figure 3. Reading tutor interface and results

Among the longer texts recorded, about one third were mail messages or personal letters, e.g.

- 1) 「今、コーヒーを飲みながらメールを書いています (^ □ ^) 」
 Ima, koohii o nominagara meeru o kaite imasu :-).
 “I am now writing e-mails while drinking coffee (smiley)”
- 2) 「レポート受け取りました。どうもお疲れ様でした。」
 Repooto uketorimashita. Doomo otsukaresama deshita.
 “I received your report. Thank you”

The rest were mostly web pages, including many wikipedia articles, Japanese articles about Slovenian companies, song lyrics, and other texts.

A surprising fact which emerged from the logs is that more than half of the “texts” which were input into Reading Tutor to be analysed were actually single words or phrases, e.g. “こんにちは” (konnichiwa “hello”), “あなた” (anata “you”), “縄文時代” (jōmon jidai “Jōmon period”), “首都” (shuto “capital”) etc. In some cases, single words were input at first e.g. “に際して” (ni sai shite “regarding”), “追っついて” (otte ite “following”), followed by longer texts of a few sentences containing these words when the user probably realised that the tool is capable of analysing and providing translations for words in longer texts.

Surprisingly many input strings (around 30%) were not Japanese character strings: most of them romanised Japanese words or phrases (e.g. “arigatou”, “kawaii”, “naruto” etc.), but also a few Slovene and English words (“ljubezen”, “love” etc.), URL addresses, and even one Chinese text. Romanized Japanese words were possibly input by users who were not able to use Japanese fonts because of their computer settings, or because they did not know enough Japanese to do so (as in the case of song lyrics with evident spelling mistakes, e.g. “kuchimoto no ugokoki [instead of: ugoki] ni yure ugoku” “wavering at lip movements”). Slovene search strings were probably input by users who overlooked the main function of the tool and used it as an online dictionary.

The frequent use of Reading tutor as a dictionary to look up single words might stem from the fact that the Slovene Reading tutor mirror is made available as a new tool alongside the online dictionary upon which it is built. Users might have had the impression that the tool with a slightly larger search box is actually only a variation or maybe newer version of the online dictionary which has been running on the same server for 5 years. A more explicit explanation of Reading tutor's functions and peculiarities is probably needed to help users better understand which of the tools is best suitable for which activity, i.e. that Reading Tutor is meant to help users read Japanese texts, while the dictionary is meant for looking up single words.

It was interesting to note also that even some users who clearly understood Reading tutor's function and input text rather than single words, still preferred to input several single sentences, which were clearly extracted from one longer text, at a few minutes intervals, rather than the whole text. One explanation for this is the users' reading habits or preferences – maybe they preferred not to rely too much on dictionaries, or preferred to read the text in its original formatting; another reason could be that they overlooked the possibility of clicking on any word in the analysed text to summon it up in the right-side vocabulary list, and therefore input shorter sentences in order to quickly scroll down the vocabulary list. This possible overlook could also be solved by a more thorough explanation of Reading tutor's functions.

4 Adding examples to jaSlo via a parallel corpus

The 3rd version of jaSlo, released in 2006, had approximately 10,000 Japanese lemmas with cca. 25,000 Slovene translational equivalents, but only 2,375 usage examples. As examples are a very useful source of information on a particular word's semantic, syntactic, collocational and pragmatic behaviour, and as some parallel texts were already available, we decided to enhance the example data-base by building and exploiting a parallel Japanese-Slovene corpus [8]. In this section we describe the methods and resources used to build the corpus, how examples were extracted to be included into the dictionary, and a short evaluation of the examples retrieved.

4.1. Corpus building

There are nowadays large amounts of parallel texts in digital form, even already aligned texts (translation memory data-bases) for combinations of major world languages, especially for those which include English, but very few Japanese-Slovene parallel texts in digital or printed form, especially texts that have been translated directly from Japanese to Slovene or vice-versa, because there were hardly any translators for this language pair up to about 10 years ago. Out of the not very numerous Japanese-Slovene translations we collected texts which can be divided into the following 4 categories: Slovene and Japanese internet culture-specific texts, which were then translated into the other language as part of students' coursework; handouts and course materials prepared by Japanese invited lecturers at the University of Ljubljana and translated into Slovenian by department staff and students; translated fiction; and selected Web pages.

The collected texts were normalised into plain text files and aligned at sentence level, and the alignment manually validated. Japanese texts were then morphologically analysed and lemmatised using Chasen [7], while the Slovene part was lemmatized using »totale« [9]. This process yielded a parallel corpus which has 4,227 translation units (sentence pairs), 109,785 Japanese tokens (morphemes) and 83,113 Slovene tokens (words).

4.2 Extracting usage examples

All lemmas included in the Japanese-Slovene dictionary were searched for in the parallel corpus, and all parallel sentences containing one of the dictionary lemmas appended to the respective lemma. 4,648 lemmas in the dictionary were thus augmented with new examples. In the case of very frequent words, only the shortest 6 examples were chosen.

In the dictionary interface, corpus examples are graphically separated from previous constructed examples

(which were already present in the dictionary), indicating to the user that they are not edited specifically for the dictionary, but rather naturally occurring examples containing the word in question, as in the following sample dictionary entry:

(tsugou) つごう 【都合】 (N)

razpoložljivost, okoliščine, pogoji, pripravnost

- 都合がいい *ustrezati*
- 都合が悪い (わるい) *ne ustrezati*
- 日曜日 (にちようび) は都合が悪い (わるい) *Ponedeljek mi ne ustreza.*

← 1. letnik, lekcija 26 NIVO 3

Korpus:

•「この不孝者めが。その方は父母が苦しんでも、その方さえ【都合】がよければ、いいと思っているのだな。」

”Kako nespoštljivo bitje! Kljub trpljenju staršev misli samo nase!” →

•3月15日に民族学博物館において予定しておりました茶道講座オープニングのレセプションは、【都合】により延期させていただきます。

Sporočamo Vam, da je zaradi bolezni otvoritev japonske čajne sobe, ki je bila načrtovana za sredo, 15. marca 2006 v Slovenskem etnografskem muzeju, preložena. →

•古代の日本列島の原住民がなつかしい理想郷として、事あるごとに思い起こす「常世の国」は、新たな支配者として権力を確立しようとするヤマト政権にとって、おそらくそれほど【都合】のよい存在ではなかったにちがいません。

Dežela Tōkoyo, ki so se je prvotni prebivalci Japonskega otočja v antiki radi spominjali, je bila najbrž neugodna za vladavino Yamato, ki je prevzela oblast kot novi gospodar. →

All corpus examples are linked to a page with information on the text where the example comes from: authors of the original text and of the translation (when known), date and place of publication or url, source language and target language in the translation pair.

4.3. Evaluation of extracted examples

Usage examples play a very important role in a learners' dictionary, since they provide implicit information on a word's semantic, syntactic, pragmatic and collocational behaviour, and as such support both passive (reading) and active (writing) use of the target language. Exposure to multiple examples of usage of the same word contribute to its better retention, and in the context of data-driven learning they form the basis of learning itself. While explanations and definitions of a word's meaning can contribute to vocabulary acquisition through deduction, usage examples are the basis for inductive acquisition of vocabulary knowledge.

Examples which are automatically extracted from a corpus do not go through the usual editorial process of dictionary entries, i.e. analysis of a corpus of examples, synthesis of the dictionary entry meaning description and editing of appropriate examples. It cannot therefore be expected, especially given the very small size of our corpus, that automatically extracted examples should cover all senses of a word or give all its most typical syntactic and collocational patterns. However, examples thus extracted were found to generally represent common collocational and syntactical patterns, and often contributed new translational equivalents for multi word units which had not been covered in the previous draft of the dictionary. Thus

for example the lemma **あわせる** (awaseru) had been translated only as »nastaviti« and »sešteti«, as in the following examples:

(constructed examples in the existing dictionary)

腕時計を駅の時計に合わせた。 *Nastavil sem ročno uro glede na uro na železniški postaji.*

2と3を合わせると5になる。 *Če seštejemo 2 in 3, dobimo 5.*

On the other hand corpus examples for the same lemma offered other translation equivalents for the units **顔をあわせる** - *srečati* - *to encounter*, **videti se** - *to meet*, and **声を合わせて歌う** *peti skupaj* - *to sing together*, as in the following examples.

(corpus examples)

顔を合わせたくなかったから。 *Nisem te hotela srečati.*

六九年の冬から七〇年の夏にかけて、彼女とは殆んど**顔を合わせ**なかった。 *Od zime do poletja sem jo komaj kdaj videl.*

私は流しをみがきながら、雄一は床をみがきながら、**声を合わせて**歌を続けた。 *Ko sem drgnila po koritu in je Juiči brisal tla, sva pela skupaj.*

Corpus examples certainly require more effort on the part of the user, who should be aware that they are not edited examples, but rather excerpts from parallel texts which have been translated in a given translational situation and may not be exact renderings of the original text, due to pragmatical or situational constraints. Indeed, some of the examples retrieved do not contain any element which could be considered as the concrete rendering of the word for which the example was extracted.

5. Conclusion and further work

The paper presented the Japanese-Slovene dictionary jaSlo, the Slovene module of the reading-support tool Reading Tutor, and an analysis of search logs in both tools. Overall both tools, Reading tutor's Slovene module and the Japanese-Slovene dictionary were found to be used quite often. An analysis of frequent searches and problems brought to light a few possible improvements, the need for a new Slovene-Japanese dictionary, and the need for more publicity among the users, who are mostly presumably students of our University.

A method for the collection of a parallel corpus and extraction of examples to be used in a learners' dictionary was also presented. The corpus collected so far was found to be useful in the sense it provided new examples to about half the entries in our dictionary, but enlarging the corpus would give a better coverage, both in terms of number of entries covered and in terms of coverage of each entry's patterns. Given a larger amount of examples for each entry, it would be useful to measure each example's level of lexical and syntactical difficulty, as proposed in [10] and [11], and of its typicality, as measured by MI score of collocational patterns with reference to a large balanced Japanese corpus [12].

Bibliography

- [1] K. Hmeljak Sangawa. (2003). Slovar japonskega jezika za slovenske študente japonščine. In *Konferenca Jezikovne tehnologije*, pp. 102-105, Ljubljana: IJS.
- [2] Erjavec, T., I. Srdanović, K. Hmeljak Sangawa. (2004). Suroveniajin nihongo gakushuusha you jisho no xmlka. *Nihongo Kyouiku Renraku Kaigi rombunshuu*, vol. 16, pp. 45-52.
- [3] Erjavec, T., K. Hmeljak Sangawa, I. Srdanović. (2006). jaSlo, a Japanese-Slovene learners' dictionary: methods for dictionary enhancement. In *Proceedings XII EURALEX international congress*, pp. 611-616. Torino: Edizioni dell'orso.
- [4] Kawamura, Y. (2000). EDR denshika jisho o katsuyou shita nihongo kyouikuyou jisho tsuru no kaihatsu. *Nihon kyouiku kougaku zasshi*. 24(Suppl.), 7-12.
- [5] Sperberg-McQueen, C., L. Burnard (ed.). Guidelines for Electronic Text Encoding and Interchange, The XML Version. The TEI Consortium, 2002. HYPERLINK "<http://www.tei-c.org/>
- [6] Japan Foundation. (2002) Japanese Language Proficiency Test: Test contents specifications. Tokyo: Bonjinsha.
- [7] Matsumoto, Y., K. Takaoka, M. Asahara. (2007). Keitaisokaiseki shisutemu ChaSen version 2.4.0 User's Manual. {<http://sourceforge.jp/projects/chasen-legacy/document/chasen-2.4.0-manual-j.pdf/ja/2/chasen-2.4.0-manual-j.pdf>}
- [8] Hmeljak Sangawa, K., T. Erjavec. (2008) A low cost approach to building a Japanese-Slovene parallel corpus. *Denshi Jōhō Tsūshin Gakkai gijutsu kenkyū hōkoku*, 2008, vol. 108, no. 50, pp. 7-10.
- [9] Erjavec, T., C. Ignat, B. Pouliquen, R. Steinberger. Massive multi-lingual corpus compilation: Acquis Communautaire and totale. In *Proceedings of the 2nd Language & Technology Conference, April 21-23, 2005, Poznan, Poland*. pp. 32-36.
- [10] Kobayashi, T., H. Oyama, K. Sakada, Y. Taniguchi, F. Ota, N. Evans, M. Asahara, Y. Matsumoto. (2007). Nihongo dokkai shien no tame no gogigoto no yourei chuushutu kinou nitsuite. In *Proceedings of the 13th Annual Meeting of the Association for Natural Language Processing*, Tokyo: Association for Natural Language Processing.
- [11] Yoshihashi, K., L. Fu, K. Nishina. (2007). Gakushuusha ni awaseta reibun hyouji tsuru. In *CASTEL-J in Hawaii 2007 Proceedings*, p. 223-226. Honolulu: University of Hawaii.
- [12] Srdanović I., T. Erjavec, A. Kilgarriff. (2008). A web corpus and word sketches for Japanese. *Shizen genjo shori*, 15(2), pp. 137-159.

Integrating the Polish language into the MULTEXT-East family: morphosyntactic specifications, converter, lexicon and corpus

Natalia Kotsyba¹, Adam Radziszewski², Ivan Derzhanski³

¹ Institute of Interdisciplinary Studies, Warsaw University

² Institute of Informatics, Wrocław University of Technology

³ Institute of Mathematics and Informatics, Bulgarian Academy of Sciences

Abstract. In this article we discuss the theoretical background, the resources employed and the process of integrating the Polish language into MULTEXT-East (version 4) including: 1) specifying a MTE-compliant tagset for it with an indication of the restrictions on combinations of attributes; 2) creating, or rather converting, a representative lexicon of word forms with tags; 3) tagging a sample text using the prepared resources.

1 Introduction

The Polish language forms, together with Kashubian and Silesian, the Lechitic subgroup of the Western group of the Slavic branch of the Indo-European language family [Ethnologue 2009]. In terms of grammar, it is a typical Slavic language. It shares with several other Slavic languages (Slovak, Upper and Lower Sorbian) a complex category of noun class including three varieties of the masculine gender (human, animal, and thing), with a peculiar subvariety of depreciative (derogative) nouns. The most unusual feature of Polish is the cliticised present tense forms of the copula along with the newly formed synthetic past tense and conditional mood of the verb, which use the cliticised copula as a subject marker.

There is no single generally accepted standard for encoding Polish corpora. The most widely used tagset for Polish is that of the Institute for Polish Language's Corpus (IPIC, <http://korpus.pl>). Other standards exist, however, such as the ones used in the PELCRA Corpus of Polish (<http://korpus.ia.uni.lodz.pl/>) or the PWN Corpus of Polish (<http://korpus.pwn.pl/>). In the National Corpus of Polish, which is currently being compiled by a consortium consisting of the contributors above (<http://nkjp.pl/>), it is anticipated that a tagset which is slightly different from IPIC [Przepiórkowski, 2009] will be employed.

The multiplicity of encoding systems makes it difficult to match existing resources for Polish and hinders the reuse of resources available for other languages and the interoperability between processing tools. Mapping them on existing international recommendations like MULTEXT could facilitate the situation.

The MULTEXT-EAST project (MTE, <http://nl.ijs.si/ME>) produced a family of morphosyntactic tagsets for various languages (primarily of Central and Eastern Europe) based on a common formalism. With the addition of Russian in 2008 [Sharoff et al., 2008] it already covers 13 languages. As it expands, it is becoming more and more diversified, from the point of view of both language typology and linguistic description. The former direction of diversification has objective reasons, the latter is due to the differences between the traditions of grammatical description in the various countries. An attempt to analyse the representation of the hitherto encoded Slavic languages in MTE and the possibility of its extension to Polish was made in [Derzhanski, Kotsyba 2009]. A number of discrepancies were identified, mostly resulting from the inconsistent use of terminology in the description of phenomena found in more than one language. What is more, some solutions already applied in MTE appear not to be extensible.

In this article we discuss the theoretical background, the resources employed and the process of integrating the Polish language into MTE including: 1) specifying a MTE-compliant tagset for it with an indication of the restrictions on combinations of attributes; 2) creating, or rather converting, a representative lexicon consisting of word forms with tags; 3) tagging a sample text basing on the prepared resources.

2 Design of the tagset

In this section the particularities of the MTE morphosyntactic specifications for Polish are explained, the new categories and their attributes with values are presented.

General considerations

Morphological tagging means endowing every word in a text with a tag identifying its grammatical form and the lemma (citation form). The grammatical form includes classificatory, inflectional and occasionally subcategorisation features.

Generally speaking, a word in this context means a graphical unit. Some special cases call for special attention: clitics that can't be conveniently treated as affixes but are written together with their hosts (the tagging process may treat them as separate words); hyphenated compounds (these may or may not be treated as a whole); 'burkinostki'¹ (forms which only occur in a certain context, essentially forming a whole with another form across blank space).

Typically forms that are superficially identical but are perceived as different in grammar get different tags (in Slavic languages such are, for example, the 2nd and 3rd person singular aorist or imperfect forms of the verb, the dative and locative singular of *a*-declension nouns). However, different uses of the same form (for instance, within analytical forms) are not normally distinguished, although this is one of the tasks of morphosyntactic tagging.

MTE is an offshoot of, and builds upon, the MULTEXT project, which was oriented primarily towards the processing of the languages of Western Europe. It recognises 14 categories, 10 of which correspond to the traditional parts of speech. A list of features is associated with each category, and a set of values with each feature. Each word form pertaining to a given category must have all features, though some values may be marked as undefined (for example, verbs normally have person, but non-finite forms do not). On the whole, MTE tagsets have tended to adhere to the national grammatical traditions. As a side effect, the same phenomena in different languages have often been treated differently, especially in the absence of a precedent in the MULTEXT languages. Contrariwise, IPIC strays away from tradition. It classifies word forms into flexemes, which correspond to parts of speech only very roughly. Characteristically, the IPIC formalism is meant expressly for Polish.

The proposed specifications are based on a modified version of the flexemic tagset developed by Marcin Woliński and Adam Przepiórkowski for IPIC, for which a converter was written to bring that categorization closer to the MTE one. Some parts of speech (*flexemes* in IPIC terminology) were decomposed into finer categories (e.g., *qubliks*—into particles, interjections and adverbs), some were presented as combinations of selected values and attributes of existing parts of speech (derogative nouns, participles, etc.).

Thus, as in the case of Russian MTE tagset [Sharoff et al. 2008], our proposal takes into account the following:

- the consistency of MTE specifications,
- the specific features of the language,
- the possibility of automatic disambiguation of feature values,
- the de-facto standard—in our case, the IPIC tagset [Wolinski, Przepiórkowski 2003].

We shall now list and briefly discuss the categories in the tagset and the associated features. The possible values of a feature are listed after its name in brackets.

¹ The term was devised by Magdalena Derwojedowa to refer to dependent words which can be encountered and identified only in a fixed combination (as *Burkina* in *Burkina Faso*).

Noun (N)

The main classificatory feature of nouns is Type (common, proper, gerund). Gerunds (*bieganie* ‘running’) are considered a Type of Nouns (strictly speaking, they are a subtype of common nouns, but are treated as a type parallel to both common and proper nouns for convenience). The features Aspect (progressive, perfective) and Negation (no, yes) are added to characterize gerunds.

The complex category of noun class that Polish shares with with Slovak and both Sorbian languages is implemented through the three features Gender (masculine, feminine, neuter), Animate (no, yes) and Human (no, yes). The values of the latter two distinguish between the masculine-human (m1), masculine-animal (m2) and masculine-thing (m3) genders of traditional grammar and of IPIC ([+Animate, +Human], [+Animate, -Human], [-Animate, -Human], respectively). This allows the relevant morphological generalisations to be captured: the feature Human is neutralised in the singular, Animate in the plural. The attribute Human also expresses what the IPIC calls derogativity (derogatives in Polish are a class of plural noun forms which are [-Human] in the nominative/vocative but [+Human] in the accusative). As both Animacy and Humanity are justified semantically and the information about them is already recorded in the morphological analyser Morfeusz², the source of grammatical information, these data are retained in the MTE tags. To technically differentiate between derogative forms of lexically [+Human] nouns and those originally marked [+Animate, -Human], the nominative/vocative plural of derogatives is encoded using the fourth theoretically possible combination, [-Animate, +Human].

The features Number (singular, plural) and Case (nominative, genitive, dative, accusative, instrumental, locative, vocative) have their traditional interpretation.

Verb (V)

Verbs are classified by Type (main, auxiliary) and Aspect (progressive, perfective).

The non-finite verb forms and the mood of the finite verb are identified by the feature VForm (indicative, imperative, infinitive, impersonal, gerund). Note that gerund as a VForm means an adverbial participle (*imięstów przysłówkowy*), not to be confused with gerund as a Noun Type (*gerundium*).

Verbs are further tagged for Tense (present, future, past). Finite verb forms have the features Person (first, second, third), Number (singular, plural), Gender (masculine, feminine, neuter) and Human (no, yes). Animacy is not relevant for verbs.

The feature Definiteness (full-art, short-art), recycled from the MTE tagset for Bulgarian, encodes here the Vocalicity of agglutinated clitics (e.g., *-em* vs *-m*). Since these are not articles, the names of the feature and both values are misnomers, but the phenomenon is similar to the Bulgarian one (essentially, allomorphy).

The feature Clitic (no, yes, agglutinant, demanding) encodes the agglutination phenomenon, which in Polish is similar to what the MTE tagset for Czech models through the feature Clitic_s for verbs and pronouns, but has a wider scope and affects more parts of speech, thus calling for a more general attribute. It is specified, e.g., for the indicative past tense form (corresponding to IPIC’s flexeme praet, the so-called pseudoparticiple) to differentiate between forms such as *gniótl* (value ‘no’) and *gniołl-* (‘demanding’), where the latter not only requires a clitic but also has different form. An ‘agglutinant’ is the clitic itself, e.g., *-em* ‘1sg’ in *gniołtem*. The value ‘yes’ is left to allow showing that a graphical word is a combination of a demanding (or free) segment and an agglutinant in case the word segmentation should be revised in the future.

No Voice feature need be defined for Polish verbs, as all verbal forms are active (adjectival/attributive participles are treated as adjectives).

² <http://nlp.ipipan.waw.pl/~wolinski/morfeusz/>

Adjective (A)

Adjectives are classified by Type (qualificative, participle). Qualificative adjectives have Degree (positive, comparative, superlative). Aspect (progressive, perfective), Voice (active, passive) and Negation (no, yes) are used for further differentiation of adjectival participles.

Gender (masculine, feminine, neuter), Animate (no, yes), Human (no, yes), Number (singular, plural) and Case (nominative, genitive, dative, accusative, instrumental, locative) work as for nouns, except that adjectives, like all other nominal categories other than nouns, have no vocative case forms.

The feature Definiteness (short-art, full-art) serves to label the IPIC flexeme *winien* ‘obliged’ and predicatives like *rad* ‘glad’ as short adjectives and to separate them from the bulk of full adjectives.

In contrast to the IPIC, ordinal numerals were extracted from adjectives and moved to numerals, and pronominal adjectives were moved to pronouns. Post-prepositional adjectives like *(po) polsku* ‘in Polish’ are treated as adverbs.

Pronoun (P)

Pronouns are subjected to the traditional classification through the feature Type (personal, demonstrative, indefinite, possessive, interrogative, relative, reflexive, negative, general). The IPIC tagset does not have pronoun types, so this information had to be supplied by hand. Further division is achieved by the features Referent_Type (personal, possessive) and Syntactic_Type (nominal, adjectival, adverbial).

Pronouns of the personal (but not the possessive) type are distinguished by Person (first, second, third). Gender (masculine, feminine, neuter), Animate (no, yes), Human (no, yes), Number (singular, plural), Case (nominative, genitive, dative, accusative, instrumental, locative) have the same interpretation as for the other nominal categories.

The feature Clitic (yes, no, agglutinant) distinguishes postprepositional forms (*nią, niego*) from regular ones (*ja, go*) and bound (agglutinating) clitics (*-ń*).³

The feature Definiteness (full-art, short-art) serves to separate full forms of pronouns (*jego, niego*) from short ones (*go, -ń*). Again, the names of the feature and both values should not to be understood literally; this attribute was used in order to avoid multiplication of attributes.

Adverb (R)

Two features are defined for adverbs: Degree (positive, comparative, superlative), as for adjectives, and Clitic (no, yes, agglutinant, burkinostka).

The IPIC tagset has a special treatment for ‘adjectival’ forms that are used to form composite adjectives (e.g., *polsko* in *polsko-ukraiński* ‘Polish–Ukrainian’). These are considered agglutinating adverbs here.

Forms which can only be used in a fixed context (e.g., *polsku* in *po polsku* ‘in Polish’) are likewise classified as special kinds of adjectives in the IPIC. In this proposal such a form is labelled as a burkinostka.

Adposition (S)

Two features are defined for adpositions: Type with a single value (preposition; there are no postpositions in Polish) and Case (genitive, dative, accusative, instrumental, locative), which encodes the preposition’s subcategorisation.

³ Cf. the value ‘bound’ of the feature Clitic for Slovene pronouns like *zame* ‘for me’ which refers to the whole cluster of a preposition and a pronoun. This coding can be used for similar phenomena in Polish, e.g., *dlań* ‘for him’, provided the word segmentation is revised towards a more traditional one.

Numeral (M)

Numerals are classified by Form (digit, roman, letter) and Type (cardinal, ordinal, collect[ive]).

Gender (masculine, feminine, neuter), Animate (no, yes), Human (no, yes), Number (singular, plural) and Case (nominative, genitive, dative, accusative, instrumental, locative) are interpreted as expected.

The feature Class (definite³⁴, definite), introduced in the MTE tagset for Czech, does what IPIC achieves through the accommodability (congr, rec) feature: ‘agreeing’ (congr) numerals such as *dwa*, *dwaj*, *trzy*, *trzej*, *cztery* have the value ‘definite 34’, whereas ‘governing’ (rec) numerals such as *pięć*, *pięciu*, *dwóch* are ‘definite’. The numeral *jeden* ‘1’ is left with the indefinite pronouns.

Particle (Q)

Particles were extracted by hand from IPIC’s *qublik* category along with adverbs, pronouns and interjections and a few conjunctions. The only feature associated with them is Clitic (no, yes, agglutinant, demanding). An agglutinant is a particle which is joined to another word (*by*, *że*). The value ‘yes’ labels a composite particle such as *niechby* when treated as one word; alternatively it may be encoded as a sequence of two particles, the optionally demanding *niech* and the agglutinant *by* (at the moment, the IPIC uses both approaches).

Conjunction (C), Interjection (I), Abbreviation (Y), Residual (X)

No features are associated with these categories.

The data associated with the proposed tagset are presented in the morphological specifications, a lexicon and a sample tagged corpus.

3 Mapping the tagsets and tags

To obtain corpora tagged with the proposed scheme, a conversion procedure was developed. It allows for conversion between the IPIC tagset and our MTE-based scheme. As the differences between tagsets are significant, the procedure is not trivial (it is discussed in the next section).

It is rather difficult to map the IPIC tagset on the MTE one without providing large lists of exceptions and conditions with lengthy explanations. Moreover, the available corpora use grammatical information coming from Morfeusz, which is not an open-source product. This is why the task of collecting the list of tags was approached empirically rather than theoretically and the mapping was basically conducted at the level of tags using information coming from already tagged corpora. For this purpose we have extracted a list of tags from the IPIC corpus.

3.1 Preparing data for conversion**3.1.1 The source corpora**

In order to extract as complete as possible a set of morphosyntactic tags for Polish we used two sources: a manually disambiguated mini-IPIC consisting of 1 mln tokens and the large IPIC itself, which amounts to approx. 250 mln tokens. The first corpus was supposed to give us relatively reliable information about the number of tags and lemmas. Theoretically, there should be no such situation when two possible disambiguations are checked manually (this happens more often in the automatically disambiguated corpus, when disambiguation criteria are not sufficient for the tagger and several options are identified as

correct).⁴ The whole IPIC has been disambiguated using an automatic tagger, therefore the tag count statistics may be biased. Nevertheless, as it is 264 times larger than the manually-disambiguated one (as measured in tokens), we decided to employ both. Surprisingly, not only do the numbers of tag types in the two corpora not coincide, but there is a large group in each that is not present in the other. This is explained in part by differences in notation between corpora. For example, *ppron12* receives the additional value of accentability in the large IPIC and this is reflected in the tags. So, both tags with and without this feature are available and used for the same forms in texts, which unnecessarily doubles their quantity.

3.1.2 Lemmatization

One of the problems of using the two corpora together as one source of information is that the lemmatization strategy differs slightly. This does not affect the list of tags but influences the lexicon and converter.

Most discrepancies in the lemmatization concern personal pronouns. In the small corpus there are three different lemmas for *ppron3* (3rd person personal pronouns): *on*, *ona*, *ono* ‘he, she, it’. In the large one they are all represented by the lexeme *on* ‘he’⁵. For the purposes of both the taglist and the lexicon all such tags were relemmatized back to the small corpus pattern with three lemmas.

Gerunds are treated differently in the two corpora: in the small one they are lemmatized as their nominative forms, in the big one as the infinitive they are derived from. For the purpose of the lexicon, as well as in the converter, the lemmas were restored to the nominative case of the noun form. Also, negation has a more morphological status in the big corpus and lemmas are presented there without the negative prefix *nie-*. This was retained in the MTE version, where nouns possess negation (because gerunds are one of the types of noun).

3.1.3 The problem of disambiguation

Some disambiguation issues had to be dealt with also in the smaller, manually disambiguated corpus. This is connected, first of all, with truly ambiguous cases, when a word and the whole phrase can be interpreted in different ways. This is unavoidable but also extremely rare. Most of the other situations concern cases where two or more IPIC tags are mapped to a single MTE tag because in IPIC personal pronouns of the first and second person are tagged for gender, or past tense masculine verb forms for animacy and humanity, which is not done in MTE. For example, the verb *dal* ‘(he) gave’ has three tags selected as correct (*praet:sg:m1:perf*, *praet:sg:m2:perf*, *praet:sg:m3:perf*), all of them corresponding to a single MTE one: *Vmeis-sm* (i.e., *Verb Type=main Aspect=perfective VForm=indicative Tense=past Number=singular Gender=male*). This naturally simplified the task of counting tags and their usage.

3.2 Conversion of tags

The collected tags amounted to 1295, including 898 tags from the small corpus and 397 tags from the big corpus that were absent in the small one. The tags were further processed and transformed into their closest MTE correspondents. They were split into their minimal values and recorded in a relational database with each value taking a separate column. Then the notation of values was replaced by the MTE one and their order was rearranged to fit the new tagset.

A large part of the original tags were mapped unconditionally. The rest had to be mapped on several MTE tags and the conditions of mapping were defined by special lists of lexemes that had to be treated as separate groups. For example, IPIC adjectives are mapped onto adjectives proper, adjectival pronouns and ordinal numerals. As the latter two are closed groups, their sets were defined in the lists of lexemes. In the

⁴ We explain such cases and their origin below.

⁵ In the MTE-3 Slavic languages whose lexicons are available for exploration there is no agreement either on how these forms should be lemmatized. Czech *my* ‘we’, *vy* ‘you (pl)’ are lemmatized as *já* ‘I’ and *ty* ‘you (sg)’, respectively. The situation in Slovene is the same. In Serbian and Bulgarian all four are different lemmas.

remaining cases a lexeme was referred to the adjectives proper.

In some cases MTE demands a more detailed description of categories than the IPIC; such divisions were introduced manually and recorded as lists of lemmas to be assigned specific tags. On the other hand, some original tags were simplified, which significantly reduced their number. The tags in the IPIC column⁶ can be divided into the following groups:

- those that are mapped to exactly one tag in the MTE map (1192 tags): comparative and superlative degree forms of adjectives, verbs, adjectival participles, gerunds, cardinal numerals, depreciative nouns, personal and reflexive pronouns, plural forms of nouns, prepositions.
- those subjected to additional division into MTE groups, first of all qubliks and non-personal pronouns.
- new tags: collective numerals, some missing pronoun forms that were deduced.
- tags that were combined into one.

We discuss some of those cases in more detail below, and the distribution of tags according to categories and source corpora is summarized in Figure 7.

3.2.2 Expanding the IPIC tags

The overall number of IPIC tags, the arithmetic sum from both corpora, that we have managed to extract amounts to 1298.⁷ 101 of them have received more than one projection in the MTE tags. Those are grouped in the following way: 60 tags for adjectives in the positive (neutral) degree of comparison were projected to 13 tags each; 18 substantive tags, to 2–7 tags each; qubliks were split into 7 categories with 27 unique tags, cf. Figure 1; predicatives were split into 3 categories with 4 tags. Such a large expansion of adjectival tags is connected first of all with separating ordinal numerals and adjectival pronouns from adjectives proper. Secondly, adjectival pronouns were split into semantic types (basically, 11 combinations of the Type and Referent_Type features in MTE), as practised in the MTE tradition. Similarly, subst tags for nouns were split into nouns proper and pro-nouns, the latter also having eight semantic types. The qublik class⁸ contained adverbs that do not inflect for degree. Those were manually marked as such and relegated to adverbs (R). Apart from this, qubliks include all interjections (I) and pronouns, mostly adverbial but also a few adjectival ones, and the short reflexive *się* (P). A few conjunctions (C) and prepositions (S) were also redirected from qubliks to corresponding classes. Figure 1 below shows the distribution of qubliks into MTE classes with number.⁹

Figure 1. Distribution of qubliks in MTE projection.

Category	Example	MTE tags	Tokens
C	alboż	1	11
I	hej	1	179
P	jakoś, się	16	85
Q	że	2	74
R	wczoraj	4	233

⁶ They cannot be called IPIC tags as some of them were added by us.

⁷ 45 tags for numerals arising from permutation of attributes but not realized in the Polish language are not included into this list. They are present, however, among the tags rejected by the TaKIPI tagger during disambiguation of corpus texts. Along with the closedness of Morfeusz this is another reason for taking tagged corpora as the starting point for extraction of tags.

⁸ The name of the category originates from the Polish word *kubło* ‘waste-paper basket’, which explains well the concept behind it.

⁹ We are thankful to the participants of the Slavic Corpora discussion group who, with their comments and advice, helped to resolve some doubtful issues concerning the division of qubliks.

Category	Example	MTE tags	Tokens
S	ponad	2	7
X	mocium	1	8

In the treatment of predicatives we followed the approach explicated in [Derzhanski, Kotsyba 2008]: copulative *to* is classified as a pronoun, the items with the morphological properties of verbs (infinitives of verbs of perception), adjectives (short forms) or nouns (citation forms) as these same parts of speech, and all others as adverbs.

3.2.3 New tags

New interpretations were added very sparingly. Figure 2 below shows two new IPIC tags (those with no entries for quantity of tokens) for short feminine forms of personal pronouns in the genitive and accusative.

Figure 2. Example of added IPIC tags and their MTE correspondents.

IPIC tag	MTE tag	MTE extended	Tokens	Example
ppron3:sg:gen:f:ter:nakc:praep	Pp-3f--sgy-n	Pronoun Type=personal Person=third Gender=feminine Number=singular Case=genitive Clitic=yes Syntactic_Type=nominal	44	<i>niej</i>
ppron3:sg:gen:f:ter:nakc:praep	Pp-3f--sgasn	Pronoun Type=personal Person=third Gender=feminine Number=singular Case=genitive Clitic=agglutinant Definiteness=short-art Syntactic_Type=nominal		<i>ń</i>
ppron3:sg:acc:f:ter:nakc:praep	Pp-3f--say-n	Pronoun Type=personal Person=third Gender=feminine Number=singular Case=accusative Clitic=yes Syntactic_Type=nominal	11	<i>nią</i>
ppron3:sg:acc:f:ter:nakc:praep	Pp-3f--saasn	Pronoun Type=personal Person=third Gender=feminine Number=singular Case=accusative Clitic=agglutinant Definiteness=short-art Syntactic_Type=nominal		<i>ń</i>

Differentiating collective numerals from cardinal ones is theoretically foreseen in the IPIC (there is a special tag for this subcategory) but not implemented in the corpus. We have added 12 new tags for such forms (masculine and neuter times six cases). Neither animacy nor humanity were relevant there. The forms are the same for the masculine and the neuter, but the gender distinction was preserved as they cannot be used with feminine nouns.

3.2.4 Collapsing the IPIC tags

Preserving all possible information was our priority, so in fact collapsing means a more economic way of recording information. This is why decisions about rejecting some tags only seemingly led to losing data, as they were superfluous in practically all cases. For example, the three masculine genders

differentiated in IPIC (m1, m2, m3) were replaced by a single masculine gender (m), but the information about peculiarities of inflexion encoded by m2 and m3, provided it is relevant in a particular case, is still stored in an MTE tag, being expressed by the categories of animacy and humanity. Numerous tags were simplified in this way in the following categories: adjectives, ordinal numerals, adjectival participles, verbal *l*-participles, numerals, and, most of all, personal pronouns.

Morfeusz presents a very detailed characteristics of word forms, often retaining attributes useless for differentiation. This leads to many tags that are never found in texts and have no theoretical justification. Moreover, they make disambiguation more difficult. For example, 3rd person personal pronouns (ppron3 flexeme in the IPIC) in general foresees 287 different IPIC tags that serve to describe 5 lemmas and their 23 forms. They are expressed by 65 MTE tags.

A similar situation is with the 1st and 2nd person personal tags (flexeme ppron12). There 146 such original IPIC tags map on 30 MTE ones.

All in all, there are 42 forms of personal pronouns in the IPIC and 433 tags for them, which were collapsed to 95 in the MTE version. The distribution of quantity of tags per word form is unequal, starting from the form *nim* with 53 interpretations in IPIC, followed by *nich* 33 and *nimi* 25 (16 forms with 10 or more interpretations) to *mu*, *jemu*, *jq* with 3 or 4 interpretations.

IPIC tags possess such attributes as accentability and prepositionality which are realized only in some forms. The extra two genders (m2 and m3) also unnecessarily increased the number of tags.

Figure 3. Tags for the 3rd person singular feminine personal pronouns' forms.

IPIC tag	MTE tag	Word form
ppron3:sg:acc:f:ter:akc:npraep	Pp-3f--san-n	<i>ja</i>
ppron3:sg:acc:f:ter:akc:praep	Pp-3f--say-n	<i>niq</i>
ppron3:sg:acc:f:ter:nakc:npraep	Pp-3f--san-n	<i>ja</i>
ppron3:sg:acc:f:ter:nakc:praep	Pp-3f--say-n	<i>niq</i>
ppron3:sg:acc:f:ter:npraep	Pp-3f--san-n	<i>ja</i>
ppron3:sg:acc:f:ter:praep	Pp-3f--say-n	<i>niq</i>

Legend:

Pp-3f--san-n: Pronoun Type=personal Person=third Gender=feminine Number=singular Case=accusative Clitic=no Syntactic_Type=nominal

Pp-3f--say-n: Pronoun Type=personal Person=third Gender=feminine Number=singular Case=accusative Clitic=yes Syntactic_Type=nominal

Figure 3 shows the situation with the two singular accusative forms of the personal pronoun *ona* 'she', which differ only in their prepositionality feature (the last two tags are from the mini-IPIC). The large IPIC adds the accentability attribute (short and full form in MTE-Polish specifications) that is not realized in the accusative, increasing the general quantity of tags to six. In the MTE tagset they were reduced again to two.

Let us have a look at some examples of disposing of the gender value in adjectivals. First the feature of gender as understood in the IPIC corpus was recast into 3 values: Gender proper, Animacy and Humanity. This gave the same number of combinations as the IPIC tagset. Further, Animacy and Humanity never have to be set simultaneously: every combination needs to contain only Gender and Humanity (66 original IPIC tags are represented by 22 MTE ones with with no Animacy value and Human=yes to differentiate between forms of nominative and accusative plural), or only Gender and Animacy (33 original IPIC tags are represented by 22 with no Humanity value and Animate=yes to differentiate between forms of accusative singular), or Gender alone. This led to a significant decrease in the number of target tags from 660 IPIC-based ones¹⁰ for adjectival pronouns to 429 MTE ones and 629 IPIC tags grouped together as

¹⁰ Originally 110 but multiplied by 6 for each semantic type.

adjectives to 425 MTE ones (including 439 active and passive adjectival participles mapped on 301 MTE ones), and finally 60 ordinal numerals split from the IPIC adjectives to 39 MTE ones.

Figure 4. Tags for ordinal numerals, the accusative case.

IPIC tag	MTE direct correspondent	MTE revised	MTE tag expanded	Example
adj:pl:acc:f:pos	Mlof--pa	Mlof--pa	Numeral Form=letter Type=ordinal Gender=feminine Number=plural Case=accusative	<i>pierwsze</i>
adj:pl:acc:m1:pos	Mlomyypa	Mlom-ypa	Numeral Form=letter Type=ordinal Gender=male Human=yes Number=singular Case=accusative	<i>pierwszych</i>
adj:pl:acc:m2:pos	Mlomynpa	Mlom-npa	Numeral Form=letter Type=ordinal Gender=male Human=no Number=singular Case=accusative	<i>pierwsze</i>
adj:pl:acc:m3:pos	Mlomnnpa	Mlom-npa	Numeral Form=letter Type=ordinal Gender=male Human=no Number=singular Case=accusative	<i>pierwsze</i>
adj:pl:acc:n:pos	Mlon--pa	Mlon--pa	Numeral Form=letter Type=ordinal Gender=neuter Number=plural Case=accusative	<i>pierwsze</i>
adj:sg:acc:f:pos	Mlof--sa	Mlof--sa	Numeral Form=letter Type=ordinal Gender=feminine Number=singular Case=accusative	<i>pierwszą</i>
adj:sg:acc:m1:pos	Mlomyysa	Mlomy-sa	Numeral Form=letter Type=ordinal Gender=male Animate=yes Number=singular Case=accusative	<i>pierwszego</i>
adj:sg:acc:m2:pos	Mlomynsa	Mlomy-sa	Numeral Form=letter Type=ordinal Gender=male Animate=yes Number=singular Case=accusative	<i>pierwszego</i>
adj:sg:acc:m3:pos	Mlomnnsa	Mlomn-sa	Numeral Form=letter Type=ordinal Gender=male Animate=no Number=singular Case=accusative	<i>pierwszy</i>
adj:sg:acc:n:pos	Mlon--sa	Mlon--sa	Numeral Form=letter Type=ordinal Gender=neuter Number=singular Case=accusative	<i>pierwsze</i>

The combinations of gender, animacy and humanity corresponding to the meanings of m1, m2 and m3 are shown in the second column. In the plural the forms *pierwszych* and *pierwsze* are differentiated only by the feature of humanity, this is why the values for animacy were removed. In the singular, the forms *pierwszego* and *pierwszy* are differentiated only by animacy, so the values for humanity were removed. This spares us 2 extra tags. Thus, only 9 out of 60 original IPIC tags retain features differentiated originally by the three masculine genders.

Another example of collapsing tags can be seen in verbal stem forms. The category of animacy was removed from this group, while humanity was left to differentiate such cases as *były* ‘were (non-m. human)’ and *byli* ‘were (m. human)’. However, this feature is important only for the plural forms. In the

singular we get 6 tags out of the original 18: the ones in figure 5 plus the same combinations for the imperfective (progressive) aspect.

Figure 5. Tags for the *l*-participle.

IPIC tag	MTE tag	MTE tag expanded	Word form
praet:sg:m1:perf	Vmeis-sm	Verb Type=main Aspect=perfective VForm=indicative Tense=past Number=singular Gender=masculine	<i>został</i>
praet:sg:m2:perf	Vmeis-sm	Verb Type=main Aspect=perfective VForm=indicative Tense=past Number=singular Gender=masculine	<i>został</i>
praet:sg:m3:perf	Vmeis-sm	Verb Type=main Aspect=perfective VForm=indicative Tense=past Number=singular Gender=masculine	<i>został</i>
praet:sg:m1:perf:agl	Vmeis-sm--d	Verb Type=main Aspect=perfective VForm=indicative Tense=past Number=singular Gender=masculine Clitic=demanding	<i>odniósł</i>
praet:sg:m2:perf:agl	Vmeis-sm--d	Verb Type=main Aspect=perfective VForm=indicative Tense=past Number=singular Gender=masculine Clitic=demanding	<i>odniósł</i>
praet:sg:m3:perf:agl	Vmeis-sm--d	Verb Type=main Aspect=perfective VForm=indicative Tense=past Number=singular Gender=masculine Clitic=demanding	<i>odniósł</i>
praet:sg:m1:perf:nagl	Vmeis-sm--n	Verb Type=main Aspect=perfective VForm=indicative Tense=past Number=singular Gender=masculine Clitic=no	<i>poniósł</i>
praet:sg:m2:perf:nagl	Vmeis-sm--n	Verb Type=main Aspect=perfective VForm=indicative Tense=past Number=singular Gender=masculine Clitic=no	<i>poniósł</i>
praet:sg:m3:perf:nagl	Vmeis-sm--n	Verb Type=main Aspect=perfective VForm=indicative Tense=past Number=singular Gender=masculine Clitic=no	<i>poniósł</i>

Figure 6 shows a very rough correspondence of categories in the MTE and IPIC.

Figure 6. Projection of MTE basic categories on IPIC ones.

MTE category	Closest IPIC flexeme
Noun (N)	subst(-) ger
Verb (V)	verb(-)*
Adjective (A)	adj(-)
Adverb (R)	adv(+)
Pronoun (P)	subst(-) adj(-)
Numeral (M)	num (+)
Particle (Q)	qub(-)
Adposition (S)	prep (-)**
Conjunction (C)	conj (+) **
Residual (X)	ign(-)
Abbreviation (Y)	ign(-)
Interjection (I)	qub(-)

Legend:

* understood as IPIC alias for verbal flexemes, without the gerund (-nie form)

** slight modifications

(+) as well as from other categories

(-) but not all of them

We tried to present the main corresponding flexeme.

We can see from the table and the legend that conjunctions and prepositions are the only parts of speech in the IPIC whose interpretation coincides with MTE. Among the few exceptions are such words as *niby, jak* 'as, like' that are classified in IPIC as prepositions governing the nominative case. They are treated as conjunctions in MTE, where the specifications for prepositions do not allow them to subcategorise for the nominative. Also, a few conjunctions were found in the *publik* class.

3.3 Statistics of tags

The quantities of tag types in the original (both IPIC corpora) and the target tagsets are very close: 1295 in the IPIC and 1266 in the MTE. Their content and informativity, however, differs greatly. (On their way to the final number, while being converted, they passed through a reduction of a nearly twice larger overall quantity.)

The MTE tag list contains 1266 tags, 102 of them have been obtained from more than one IPIC tag.

Figure 7. Correspondence of tags depending on the category and the source corpus.

	Original IPIC tags (M)*	Original IPIC tags (A) **	Expanded IPIC tags (M)	Collapsed IPIC tags (M) MTE
Noun (N)			95	95
subst	69			71
depr	2			
ger	21	3	24	24
Verb (V)		-	71	56
aglut	6			
bedzie	6			
fñ	12			
imps	2			
impt	6			
inf	2			
praet	32			
Adjective (A)		203	629	425
adj	171	11 (comp/sup degree)		

pact	82	125		
ppas	167	65		
pcon	1	1		
pant	1	1		
winien	10	-		
Adverb (R)	3		7	6
pred	1			
Pronoun (P)		182 (only personal)	1167 (with new ones)	597
ppron12	140 all 146-443		146	30
ppron3	107 all 287-443		287	65
siebie	5		5	5
Numeral (M)			114	75
cardinal	33	3 ¹¹	34 (transfer from adj)	22
ordinal			60 + 2	39
collective			18 (newly added)	12
Particle (Q)	1	-	1	1
Adposition (S)	14	-	15 (transfer from qub)	6
Conjunction (C)	1	-	1	1
Residual (X)	1	5 ¹²	8	1
Abbreviation (Y)	-	-	1	1
Interjection (I)	1	-	1	1
Total	898	397	2157 (without 45 theoretically impossible)	1266

* M – manually disambiguated corpus

** A – automatically disambiguated corpus, only the new tags that were absent from M.

3.4 Word segmentation

One of the major differences between the IPIC approach and the MTE one is in the word segmentation principles. This is not a trivial issue and calls for the development of an optimal strategy for dealing with such situations in the future. The IPIC approach is a highly practical and economic one but it deviates from the traditional understanding of what a word is, which is realized in the MTE records of language material. A typical example of token representation in the IPIC:¹³

```
<orth>mogli</orth><lex disamb="1"><base>móc</base><ctag>praet:pl:m1:imperf</ctag></lex>
<ns/>
<orth>by</orth><lex disamb="1"><base>by</base><ctag>qub</ctag></lex>
<ns/>
<orth>ście</orth><lex disamb="1"><base>być</base><ctag>aglt:pl:sec:imperf:nwok</ctag></lex>
```

Here one graphical word *moglibyście* ‘you(pl) could’ is presented by three segments with their own lemmas. The same word in the MTE notation (before revising its segmentation):

```
<w lemma="móc" ana="Vmpis-pmy">mogli</w>
<w lemma="by" ana="Q">by</w>
<w lemma="być" ana="Vapip2p--sa">ście</w>
```

Legend:

Vmpis-pmy: Verb Type=main Aspect=progressive VForm=indicative Tense=past Number=plural
Gender=male Human=yes

¹¹ Including two tags for digits added by the TaKIPI tagger, whereas in IPIC digits would be classified as residuals (ign).

¹² As in the case with numerals, these are the TaKIPI tagger tags that are “ignorable” for both the IPIC and the MTE. Examples: tdate, tmail, turi, tdate, tsym.

¹³ The <tok> tags were removed here to simplify the representation.

Q: Particle

Vapip2p--sa: Verb Type=auxiliary Aspect=progressive VForm=indicative Tense=present Person=second Number=plural Definiteness=short-art Clitic=agglutinant

The IPIC notation includes a “no space” tag <ns/> to signal cases when a segment of a word is presented as a separate lemma in the corpus. This allows several problems to be solved: the floating ending of the past indicative verb forms (a remnant from the old analytical perfective form) which can be attached practically to everything (nouns: *swiniaś* (*świnia jesteś*) ‘pig (you) are’, pronouns: *tyś* (*ty jesteś*) ‘you are’, conjunctions: *żebyście* (technically: *że by jesteście*) ‘in order for you(pl) to (be)’, adverbs: *wcaleś* (technically: *wcale jesteś*) ‘at all (you) are’, etc.) and the multiplication of verbal forms that can be created according to strict agglutinating rules: *myślał-by-m* ‘I would think’, *znalazł-by-ś* ‘you would find’. If we wanted to treat all such clusters as single words, we would frequently be at a loss for a way to name them or would have to introduce a bulky category of predicativity for nouns, adverbs, etc., and further complicate the interpretation of their morphology. These cases are treated as technically combined independent words. Combinations of prepositions and pronouns like *dlań* (*dla niego*) ‘for him’ are marked in the MTE tagset with the help of the Clitic feature for pronouns. The value a(gglutinant) shows that the string is technically part of an orthographic word, cf. Figure 8.

Figure 8. Morphological tagging for strings like *dlań*.

dlań	dla	Spg	Adposition Type=preposition Case=genitive
	ń	Pp-3m--sgasn	Pronoun Type=personal Person=third Gender=male Number=singular Case=genitive Clitic=agglutinant Definiteness=short-art Syntactic_Type=nominal

This, however, means that each segment receives an independent morphosyntactic interpretation, including tense etc. information (cf. the interpretation of *moglibyście* above), which is at variance with traditional grammatical description and speakers’ intuitions. We believe that the problem can be solved and a more truthful picture can be achieved by the partial use of a secondary grouping. However, not all of these cases can and need to be treated as whole words (let us remember that orthographic rules are often a matter of convention).

We will distinguish cases when the agglutinant rambles away (*bym mógł, swiniaś, dlań*) and when it accompanies its master participle. The former will have to await further analysis using syntactic parsing, as it is not always possible to technically differentiate between situations when it is originally an ending of the past verbal form that carries the information about the category of person and when it represents an independent verb in present tense. The latter was modified by combining both segments’ forms and their grammatical information to generate a single tag for the whole.

Thus a two-segment word *mogliście* after revising its segmentation looks in the MTE notation as follows (cf. with a three-segment word above):

<w lemma="móc" ana="Vmpis2pmy-y">mogliście</w>

A similar situation obtains with the clitic *-by*, which introduces the conditional mood. This clitic can be a standalone word form (when it precedes the verb) or a part of the verb form. In the latter case, the verb stem and the clitic are combined into a single token with a new grammatical information. The Tense value is changed into “present” and the Form acquires the value “conditional” instead of the former “indicative”. As well as in the example above, the clitic can also be followed by a floating ending—in such cases all the information is integrated into a single verb token.

Below are two examples of conversion: a third person plural conditional verb form, *mogliby* ‘they could’, and a second person plural conditional verb form, *moglibyście* ‘you(pl) could’.

<w lemma="móc" ana="Vmpec3pmy-y">mogliby</w>

```
<w lemma="móc" ana="Vmpcp2pmy-y">moglibyście</w>
```

3.5 Tag converter

The discussed conversion method has been implemented in the Python programming language. The converter consists of source code and separate files with conversion tables (tab-delimited lists). Each entry in the main conversion table may be a 1:1 tag correspondence or a reference to another conversion table (for lemma-based conversion rules). When running, all the tables are first read and indexed, which allows for faster performance. The converter reads IPIC XML files and produces TEI XML output compliant with other MTE sample corpus files.

As noted above, the conversion is conducted at the level of tags, i.e., the conversion tables provide a closed list of tags and rules for their conversion, with no generalisation. The obvious disadvantage is that we may encounter an unexpected tag. This solution still seemed preferable since it is not an easy task to capture a reasonable generalisation within a moderate set of rules while assuming that the employed list of tags is quite extensive. What is more, some well-formed IPIC tags are practically impossible, if not invalid—it may be desirable to get explicit information about such cases. The out-of-list tags are converted to residuals (X) and reported to the user.

4 Deliverables

In order to include a new language into MTE, the following package should be prepared: morphosyntactic specifications with a MSD index (representative list of possible tags), a lexicon and a sample of a tagged corpus.

4.1 Morphosyntactic specifications

The morphosyntactic specifications have been prepared in TEI XML format. The whole description is contained within one XML file with several sections. The file commences with a header containing metadata, followed by the main part which specifies each category, its attributes and their possible values. Every category is followed by optional notes/comments and a table which presents possible combinations of tags for this particular category. XML files can be transformed into HTML format, which is more convenient for the human reader, with the help of special XSLT writing scripts (stylesheets) provided by MTE V.4 developers, cf. [Erjavec 2009].

Figure 9 shows a fragment of the specifications as they look in HTML format (Polish adverb).

Figure 9. A fragment of the specifications in HTML (Polish adverb)

0 CATEGORY	Adverb	R
1 Degree	positive	p
	comparative	c
	superlative	s
2 Clitic	yes	y
	no	n
	agglutinant	a
	burkinostka	u

The last part of the specifications – the MSD index – consists of an extensive tag list, providing token occurrence count as well as example forms and lemmas. Both source corpora were fed through the converter. Employing both of them was significant, since there is a slight difference in the adopted tagging

scheme: some categories are considered optional and omitted in the smaller corpus (we wanted to acquire all of the allowed tags). The resulting lists of tags are combined; the overlapping part is taken from the manually disambiguated corpus. To balance the acquired tag occurrence counts, we multiply the counts taken from the bigger corpus by an appropriate ratio.

Figure 10. A fragment of the MSD index.

MTE tag	MTE expanded	Types	Example
Vmeis2sf--y	Verb Type=main Aspect=perfective VForm=indicative Tense=past Person=second Number=singular Gender=feminine Clitic=yes	85	<i>powiedziałaś/powiedzieć, zrobiłaś/zrobić, przyszłaś/przyjść</i>
Vmeis2sm--y	Verb Type=main Aspect=perfective VForm=indicative Tense=past Person=second Number=singular Gender=masculine Clitic=yes	274	<i>przyszedeś/przyjść, powiedziałeś/powiedzieć, zrobileś/zrobić,</i>
Vmeis2sn--y	Verb Type=main Aspect=perfective VForm=indicative Tense=past Person=second Number=singular Gender=neuter Clitic=yes	1	<i>pozostałoś/pozostać, przeszłoś/przejsć</i>
Vmeis-pf	Verb Type=main Aspect=perfective VForm=indicative Tense=past Person=second Number=plural	619	<i>odbyły/odbyć, rozpoczęły/rozpocząć, zaszły/ zajść</i>

Whenever possible, three examples of form/lemma pairs for a tag (some tags occur with one or two distinct forms only) are provided. To lower the number of repetitions, a simple heuristic for the selection of examples was employed. Some tokens in the corpus contain more than one candidate tag. Fortunately, many of these ambiguities disappeared after the conversion (as the proposed standard does not follow all the distinctions introduced in IPIC, which was a major cause of insoluble ambiguities). Nevertheless, some of them remained, resulting in troublesome situations, especially those coming from the big corpus submitted to an automatic disambiguation. We decided to count such candidate tags as fractions of occurrences (their counts adding up to 1 for a token).

4.2 The lexicon

The lexicon is meant to provide full inflection paradigms of the most frequent lemmas. As no extensive lexicographic resource with such information is available for Polish, we resorted to the corpus (IPIC). The 15 thousand most frequent lemmas were extracted from it with the help of PoliQarp.¹⁴ Then the remaining forms for those lemmas were extracted from the large corpus. The lexicon includes a word form, its lemma, its tag and the number of token occurrences in the IPIC.

Figure 6. A fragment of the lexicon.

absurdami	absurd	N-mnnpj	17
absurdem	absurd	N-mnnsi	307
absurdom	absurd	N-mnnpd	6
absurdowi	absurd	N-mnnsd	4
absurdu	absurd	N-mnmsg	578
absurdy	absurd	N-mnnpa	59

¹⁴ <http://korpus.pl/index.php?page=poliQarp>

absurdy	absurd	N-mnnpn	58
absurdzie	absurd	N-mnnsł	17
absurdów	absurd	N-mnnpğ	163
aby	aby	C	201168
ac	ac	X	1099
ach	ach	I	1170

The total number of unique word forms in the lexicon is 175848 (roughly 11.72 per lemma), while the number of forms with all possible interpretations is 339031.

4.3 The corpus

The MTE-like tagged corpus in our case consists of one book, approx. 100000 words, namely George Orwell's *1984*. This book was chosen because it was used for the MTE multilingual parallel corpus for 11 languages, thus adding it was a natural way to extend the multilingual MTE parallel corpus for Polish and is intended to facilitate the validation of the specifications for Polish and the converter on sufficiently large language data.

The tagging was performed with the help of TaKIPI program, cf. [Broda et al. 2008], specially developed for tagging Polish using IPIC tagset. Afterwards the tag converter was used to bring it to MTE-style format. The resulting corpus contains 79807 word tokens and 17642 punctuation mark occurrences. The word tokens appear with 801 different MTE tags and 9480 different lemmas. Below we present a fragment of the corpus in the TEI XML format:

```
<p id="Opl.5">
<s id="Opl.5.1">
<w lemma="być" ana="Vmpis-sm">Był</w>
<w lemma="jasny" ana="A-pm--sn">jasny</w>
<c>,</c>
<w lemma="zimny" ana="A-pm--sn">zimny</w>
<w lemma="dzień" ana="N-mnnsa">dzień</w>
<w lemma="kwietniowy" ana="A-pmn-sa">kwietniowy</w>
<w lemma="i" ana="C">i</w>
<w lemma="zegar" ana="N-mnnpn">zegary</w>
<w lemma="bić" ana="Vmpis-pmn">biły</w>
<w lemma="trzynasty" ana="Mlof--si">trzynastą</w>
<c>.</c>
</s>
```

5 Conclusions and future work

An MTE-4 compliant package for the Polish language was prepared on the basis of existing resources and presented in this paper. This is an important step in integrating linguistic resources of Slavic languages, as it makes Polish much more comparable than it was before. Of course, this is only a first step and much remains to be done.

One point that received relatively little attention in [Derzhanski, Kotsyba 2009], but may be very important for comparative studies based on the common tagset and the parallel corpus, is that certain categories (or rather subcategories) existing in most MTE languages are only explicated in some of them. For example, the Russian MTE tagset introduces non-specific pronouns (*весь* 'all', *всякий* 'any, every', *сам* 'oneself', *самый* 'the very', *каждый* 'every, each', *иной* 'other', *любой* 'any', *другой* 'other'). This category, inspired by MAK Halliday's works, is not part of either Russian traditional grammar (the standard description of which is the Academic grammar), the theoretical premises of the Russian National Corpus, or the descriptions of other MTE languages. Nevertheless, items semantically and etymologically corresponding to the words in this group exist in all MTE Slavic languages, though classified as other types of pronouns or even other parts of speech. This issue deserves a separate investigation; here we just

want to signal that both those who deal with language description and with searching through the parallel corpus have to be aware of different granulation levels for some grammatical categories. Likewise, participles are treated variously as adjectives or verb forms in MTE lexicons. The earlier mentioned lemmatization discrepancies need to be removed. And so on.

As for Polish itself, its specific word segmentation regarding clitics needs further syntactic analysis to correct grammatical information provided by tags about some agglutinated forms of *być* ‘to be’.¹⁵ Similarly, clustering analytical verb forms for Polish and other languages would give us a picture much closer to the traditional understanding of grammar and would facilitate further linguistic research and information retrieval.

All the described resources are very “fresh” and need validation to eliminate possible mistakes. It would be very useful if online search in the existing parallel corpus were provided. Presently, the resources from MTE-3 version are available for download upon registration. However, the absence of search tools does not allow linguists to use their full capacity. We would expect that giving such a possibility to a greater public could result in a feedback from which the general quality of corpora and the rest of the resources could only benefit.

Bibliography

- [1] Broda B., Piasecki M. and Radziszewski A. (2008). Towards a Set of General Purpose Morphosyntactic Tools for Polish. In *Proceedings of Intelligent Information Systems, Zakopane, Poland, 2008*. Institute of Computer Science—PAŚ.
- [2] Derzhanski I. and Kotsyba N. (2008). The category of predicatives in the light of the consistent morphosyntactic tagging of Slavic languages. In *Proceedings of Lexicographic Tools and Techniques: Proceedings of the MONDILEX First Open Workshop*, pages 68–79, Moscow: IITP—RAS.
- [3] Derzhanski I. and Kotsyba N. (2009). Towards a Consistent Morphological Tagset for Slavic Languages: Extending MULTEXT-East for Polish, Ukrainian and Belarusian. In *Garabik 2009*, pp. 9–26.
- [4] Dimitrova L., Erjavec T., Ide N., Kaalep H.-J., Petkevič V., Tufiş D. (1998). Multext-East: Parallel and Comparable Corpora and Lexicons for Six Central and Eastern European Languages. In *Proceedings of COLING—ACL’98*, pages 315–319, Montréal, Québec, Canada.
- [5] Erjavec, T. (ed.) (2004). *MULTEXT-East Morphosyntactic Specifications: Version 3.0*. Ljubljana.
- [6] Erjavec, T. (2009). *MULTEXT-East Morphosyntactic Specifications: Towards Version 4*. In *Garabik 2009*, pp. 59–70.
- [7] Garabik, R. (ed.) (2009). *Proceedings of Metalanguage and Encoding Scheme Design for Digital Lexicography: MONDILEX Third Open Workshop*, Bratislava, 15–16 April 2009.
- [8] Kotsyba N., Shypnivska O. and Turska M. (2008). Linguistic principles of organizing a common morphological tagset for PolUKR (Polish-Ukrainian Parallel Corpus). In *Proceedings of Intelligent Information Systems, Zakopane, Poland, 2008*. Institute of Computer Science—PAŚ.
- [9] Lewis M. P. (ed.), 2009. *Ethnologue: Languages of the World*, Sixteenth edition. Dallas, Tex.: SIL International. Online version: <<http://www.ethnologue.com/>>.
- [10] Przepiórkowski A. and Woliński M. (2003). A Flexemic Tagset for Polish. In *Proceedings of the Workshop on Morphological Processing of Slavic Languages, EACL 2003*.
- [11] Przepiórkowski A. (2009). A comparison of two morphosyntactic tagsets of Polish. Version of 15 July 2009. To appear in *the proceedings of the MONDILEX workshop held in Warsaw, 29–30 June 2009*. URL: <<http://nlp.ipipan.waw.pl/~adamp/Papers/2009-mondilex/>>.
- [12] Roszko, R. (2009). Morphosyntactic Specifications for Polish. Theoretical Foundations. Description of Morphosyntactic Markers for Polish Nouns within MULTEXT-East Morphosyntactic Specifications (Version 3.0 May 10th, 2004). In *Garabik 2009*, pp. 140–150.

¹⁵ If the agglutinant is a floating ending and it is possible to identify its master, the information about the person should be added to the verbal form. Depending on whether *być* is an independent verb or the verbal ending, the same form carries different grammatical information. If it is independent the interpretation in the tag gives the truthful picture about its grammar but if it is an ending, its grammatical information is in conflict with the one of the master verbal form.

- [13] Sauvet G., Włodarczyk A. and Włodarczyk H. (2007). Morphological data exploration using the Semana platform: Feature granularity problem in the definition of Polish gender. Lecture slides: <http://www.celta.paris-sorbonne.fr/anasem/papers/miscelanea/PolishGender.pps>.
- [14] Sharoff S., Kopotev M., Erjavec T., Feldman A., and Divjak D. (2008). Designing and evaluating a Russian tagset. In *Sixth International Conference on Language Resources and Evaluation, LREC'08*, Paris, ELRA.
- [15] Woliński, M. (2004). System znaczników morfosyntaktycznych w korpusie IPI PAN. *Polonica XII*, 39–54.
- [16] Николаева Т. М. (2008). Непарадигматическая лингвистика. *История «блуждающих частиц»*. Москва. *Studia Philologica*.

Adding Multi-Word Expressions to sloWNet

Špela Vintar, Darja Fišer

Dept. of Translation, Faculty of Arts, University of Ljubljana
Aškerčeva 2, SI-1000 Ljubljana, Slovenia
spela.vintar@guest.arnes.si, darja.fiser@guest.arnes.si

Abstract. The paper presents an approach to expand Slovene wordnet with domain-specific multi-word terms by exploiting multiple resources. A large monolingual Slovene corpus of texts from the domain of informatics was used to harvest terminology from, and a parallel English-Slovene corpus and an online dictionary as bilingual resources to help with the mapping of terms to the Slovene wordnet. First, core terms of the domain were identified in English using the Princeton WordNet, which were then translated into Slovene with a bilingual lexicon produced from the parallel corpus. Next, multi-word terms were extracted from the Slovene domain-specific corpus using a hybrid statistical / pattern-based approach, and finally the term candidates were matched to existing wordnet synsets. This procedure is based on the premise that the term's headword is the hypernym of the multi-word phrase. When the headword is ambiguous, the semantic information provided in wordnet was used to choose the right sense. However, there are still many cases in which disambiguation nevertheless needs to be performed manually. The method proposed in this paper appears to be a successful way to improve the domain coverage of wordnet as it takes advantage of various multilingual resources and yields numerous term candidates.

1 Introduction

WordNet [7] is an extensive lexical database in which words are divided by part of speech and organized into a hierarchy of nodes. Each node represents a concept and words denoting the same concept are grouped into a synset with a unique id (e.g. ENG20-02853224-n: {car, auto, automobile, machine, motorcar}). Concepts are defined by a short gloss (e.g. 4-wheeled motor vehicle; usually propelled by an internal combustion engine) and are also linked to other relevant synsets in the database (e.g. hypernym: {motor vehicle, automotive vehicle}, hyponym: {cab, hack, taxi, taxicab}). Over time, WordNet has become one of the most valuable resources for a wide range of NLP applications, which initiated the development of wordnets for many other languages as well¹.

One of such enterprises is the building of Slovene wordnet [5,8,9]. While this task would normally involve substantial manual labour and the efforts of several linguists, Slovene wordnet was built almost single-handedly exploiting multiple multilingual resources including a bilingual dictionary, multilingual parallel corpora and semantically structured resources such as Eurovoc and Wikipedia. The combination of these approaches yielded the first version of the Slovene WordNet² (sloWNet) containing over 17,000 synsets and 20,000 literals. The majority of these literals are however single-word items, because the main lexicon extraction procedures involved in the building of WordNet involved no systematic handling of multi-word expressions. Since the latter constitute an important part of the lexicon, especially in specialized discourse, the purpose of this paper is to propose a method to enrich wordnet with domain-specific multi-word expressions.

The rest of the paper is organized as follows: first, the Slovene WordNet Project is described. Section 3 describes the procedure used to extract multi-word expressions from the corpus and their mapping to the wordnet hierarchy. The results are presented and evaluated in Section 4, and the paper ends with concluding thoughts and plans for future work.

2 Building the Slovene Wordnet

The first version of the Slovene wordnet was created on the basis of the Serbian wordnet [11], which was translated into Slovene with a Serbian-Slovene dictionary. The main advantages of this approach were the direct mapping of the obtained synsets to wordnets in other languages and the density of the created network.

¹ See http://www.globalwordnet.org/gwa/wordnet_table.htm [15.03.2008]

² sloWNet is distributed under the Creative Commons licence, <http://nl.ijs.si/sloWNet/>

The main disadvantage was the inadequate disambiguation of polysemous words, therefore requiring extensive manual editing of the results. The core Slovene wordnet contains 4,688 synsets, all from Base Concept Sets 1 and 2.

In the process of extending the core Slovene wordnet we tried to leverage the resources we had available, which are mainly corpora. Based on the assumption that translations are a plausible source of semantics we used multilingual parallel corpora such as the Multext-East [6] and the JRC-Acquis corpus [13] to extract semantically relevant information [8].

We assumed that the multilingual alignment based approach can either convey sense distinctions of a polysemous source word or yield synonym sets based on the following criteria (cf. [2,4]):

- (a) senses of ambiguous words in one language are often translated into distinct words in another language (e.g. Slovene equivalent for the English word '*school*' meaning educational institution is '*šola*' and '*jata*' for a large group of fish);
- (b) if two or more words are translated into the same word in another language, then they often share some element of meaning (e.g. the English word '*boy*' meaning a young male person can be translated into Slovene as either '*fant*' or '*deček*').

In the experiment, corpora for up to five languages (English, Slovene, Czech, Bulgarian and Romanian) were word-aligned with Uplug [14] used to generate a multilingual lexicon that contained all translation variants found in the corpus. The lexicon was then compared to the existing wordnets in other languages. For English, the Princeton WordNet [7] was used while for Czech, Romanian and Bulgarian, wordnets developed in the BalkaNet project [16] were used. If a match between the lexicon and wordnets across all the languages was found, the Slovene translation was assigned the appropriate synset id. In the end, all the Slovene words sharing the same synset ids were grouped into a synset.

The results obtained in the experiment were evaluated automatically against a manually created gold standard. A sample of the generated synsets was also checked by hand. The results were encouraging, especially for nouns with f-measure ranging between 69 and 81%, depending on the datasets and settings used in the experiment. However, the approach had two serious limitations: first, the automatically generated network contains gaps in the hierarchy where no match was found between the lexicon and the existing wordnets, and second, the alignment was limited to single-word literals, thus leaving out all the multi-word expressions.

We tried to overcome this shortcoming with extensive freely available multilingual resources, such as Wikipedia and Eurovoc. These resources are rich in specialized terms, most of which are multi-word. Since specialized terminology is typically monosemous, a bilingual approach sufficed to translate monosemous literals from PWN 2.0 into Slovene. A bilingual lexicon was extracted from Wikipedia, Wiktionary and Wikispecies by following inter-lingual links that relate two articles on the same topic in Slovene and English. We improved and extended this lexicon with a simple analysis of article bodies (capitalization, synonyms extraction, preliminary extraction of definitions). In addition we extracted a bilingual lexicon from Eurovoc, a multilingual thesaurus that is used for classification of EU documents. This procedure yielded 12.840 synsets. Translations of the monosemous literals are very accurate and include many multi-word expressions, and thus neatly complement the previous alignment approach. Also, they mostly contain specific, non-core vocabulary.

3 Multi-word expressions and wordnet

Multi-word expressions (MWE) are lexical units that include a range of linguistic phenomena, such as nominal compounds (e.g. *blood vessel*), phrasal verbs (e.g. *put up*), adverbial and prepositional locutions (e.g. *on purpose*, *in front of*) and other institutionalized phrases (e.g. *de facto*). MWEs constitute a substantial part of the lexicon, since they express ideas and concepts that cannot be compressed into a single word. Moreover, they are frequently used to designate complex or novel concepts. As can be seen in Table 2, the majority of MWEs in Princeton Wordnet do not belong into any of the Basic Concept Sets,

meaning that they encode specialized concepts and are frequently terms.

As a consequence, their inclusion into wordnet is of crucial importance, because any kind of semantic application without appropriate handling of MWEs is severely limited.

For the purpose of MWE identification, various syntactical [1], statistical [15] and hybrid semantic-syntactic-statistical methodologies [12,3] have been proposed, to name but a few. Since the majority of MWEs included in the Princeton WordNet are nominal (see Table 1 below) and compositional, our approach is based on syntactic features of MWEs.

POS	Freq.
nouns	60,931
verbs	4,315
adverbs	955
adjectives	739
total	66,940

Table 1: The distribution of MWEs in PWN across part-of-speech

Group	Freq.
other	64,205
BCS 3	1,470
BCS 2	926
BCS 1	339
total	66,940

Table 2: The distribution of MWEs in PWN across BCS

In addressing the issue of MWEs in sloWNet, we initially wanted to find Slovene equivalents for the MWEs already present in Princeton Wordnet. We describe this experiment and its successful implementation in [18].

However, if wordnet is to be used in a semantic application within a specific domain, we wish to ensure its coverage within this domain primarily for the target language. The goal we address here is thus how to enrich sloWNet with domain-specific Slovene MWEs regardless of whether their English counterparts are included in PWN or not.

The resources we use to this end are the following (Figure 1):

- Ikorpus, a Slovene corpus of Computer Science texts, size ca. 15 million words, morphosyntactically annotated and lemmatized,
- a Slovene-English parallel corpus of Computer Science abstracts, size ca. 300,000 words, morphosyntactically annotated and lemmatized,
- Islovar, a Slovene-English online dictionary of Computer Science³,
- Princeton WordNet.

³ <http://www.islovar.org/>

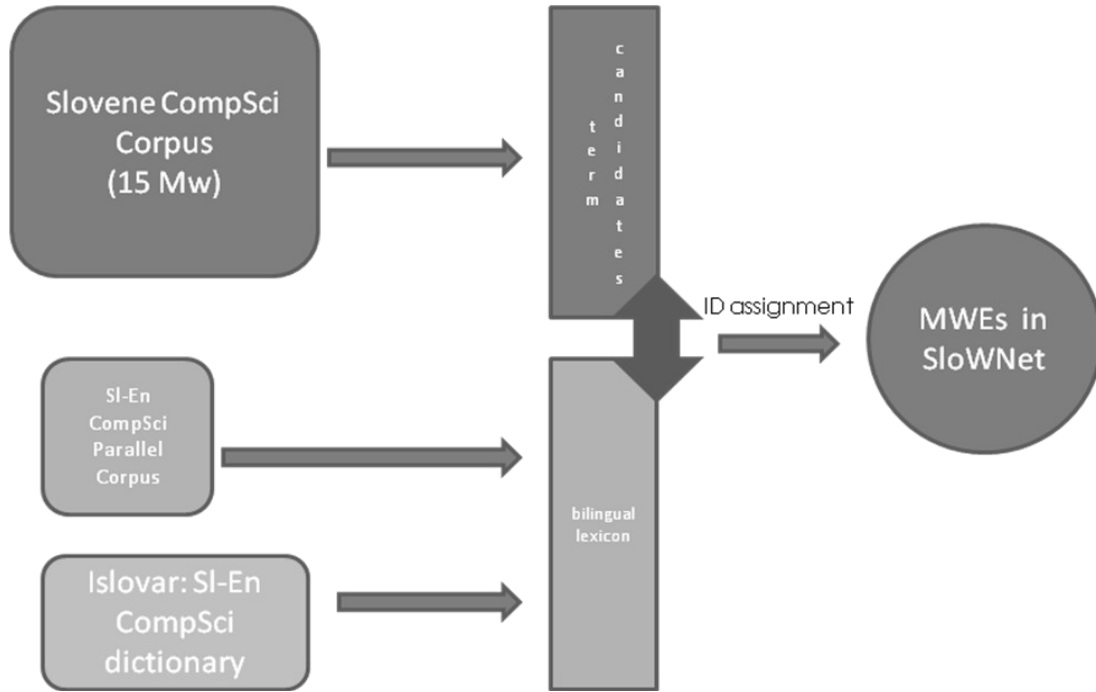


Figure 1: Resources for harvesting MWEs

3 Automatic Term Extraction

The domain-specific Ikorpus is composed of texts from 5 journals dealing with computer science, information and communication technology, and it also contains 5 consecutive volumes of proceedings of the largest informatics conference in Slovenia DSI.

Automatic term extraction from the corpus is performed using a hybrid approach based on morphosyntactic patterns for Slovene and statistical ranking of candidates [17]. The patterns, such as Adjective+Noun or Noun+Noun[Gen], yield numerous potential MWEs. To determine their terminological relevance, each MWE is assigned a weight according to its frequency and the keywordness of each of its constituent words.

$$W(a) = \frac{\sum \log \frac{f_{n,D}}{N_D} \frac{f_{n,R}}{N_R}}{n} * f_a^2$$

The term extraction procedure performed on the 15-million-token Slovene corpus of computer science yielded over 70,000 term candidates (Table 3). Since the extractor uses morphosyntactic patterns, each multi-word term candidate [e.g. *domenski strežnik* (*domain name server*)] is automatically assigned a headword [*strežnik* (*server*)] and we assume this to be the hyponym of the term candidate.

Clearly, the domain-specific terms constitute a valuable lexical resource, but not until we can introduce some semantic structure. The next step therefore is to integrate at least some of these terms into the Slovene wordnet.

MWE size	Number of candidates
2 words (Adj+N, N+N, ...)	54,844
3 words (Adj + Adj + N, N + Prep + N, ...)	16,861
4 words (Adj + Adj + N + N, ...)	2,605
Total	74,310

Table 3: Term candidates and their length in words

4 Bilingual Lexicon Extraction

At this point we have a large number of Slovene multi-word terms without any semantic information other than the headword of each unit. Thus, for a term such as *prosto programje* [*free software*], since it has been extracted through the syntactic pattern Adjective + Noun, we know that *programje* is the headword and *prosto* the modifier. We may also assume that *programje* [*software*] is the hypernym of *prosto programje* [*free software*], and hence we could add *prosto programje* [*free software*] into Slovene wordnet as the hyponym of *programje* [*software*], but only if the Slovene wordnet already has the required headword.

For many multi-word terms this turns out not to be the case, which is why we wish to add both the hypernym and its extracted hyponyms to sloWNet. We use the Princeton Wordnet as the source of semantic structure, and to be able to link headwords to PWN we use bilingual lexicon extraction.

A small English-Slovene parallel corpus of 300,000 tokens was fed to the Uplug word aligner [14], which produced suggested translations for each word found in the corpus. To improve accuracy, we use only alignments of words that occur more than once and alignment scores over 0.05. This yields a bilingual single-word lexicon of 1326 words, mostly nouns, as shown in Table 4.

Freq	Score	English	POS	Slovene	POS
4	0.058264988	active	a	aktiven	a
8	0.100445189	activity	n	aktivnost	n
5	0.138443460	agent	n	agent	n

Table 4: Sample entries in the bilingual lexicon

In order to improve coverage and accuracy, the automatically extracted bilingual lexicon was further enlarged with entries from the English-Slovene online dictionary of computer science. The dictionary was consulted also in certain cases of ambiguous headword, see following section.

5 Adding Terms to sloWNet

For each Slovene multi-word term candidate we first identify its headword and assume that the headword is its hypernym. Using our bilingual lexicon we translate the headword into English and retrieve its synset IDs from PWN. If the headword turns out to be monosemous, the entire term group can be added to the Slovene wordnet under the unique synset ID (Table 5).

Term candidates	Slo. & Eng. hypernym possible synset IDs	Selected synset ID
prosto programje [free software] priloženo programje [attached software] ustrezno programje [appropriate software] novejše programje [updated software] dodatno programje [additional software] vohunsko programje [spyware]	Programje = software ENG20-06162514-n [computer_science]	ENG20-06162514-n

Table 5: Monosemous headword

If the headword could be assigned several possible senses and one of the senses belongs to the domain Computer Science, than this sense is chosen (Table 6).

Term candidates	Slo. & Eng. hypernym possible synset IDs	Selected synset ID
vgrajena tipkovnica [built-in keyboard] brezžična tipkovnica [wireless keyboard] zaslonska tipkovnica [monitor keyboard] tipkovnica qwerty [QWERTY keyboard] navidezna tipkovnica [virtual keyboard] miniaturna tipkovnica [miniature keyboard] zunanja tipkovnica [external keyboard] zložljiva tipkovnica [folding keyboard] ergonomska tipkovnica [ergonomic keyboard] programska tipkovnica [program keyboard] slovenska tipkovnica [Slovene keyboard] modularna tipkovnica [modular keyboard] alfanumerična tipkovnica [alphanumeric keyboard]	tipkovnica = keyboard ENG20-03480332-n [computer_science] ENG20-03480198-n [factotum]	ENG20-03480332-n

Table 6: Polysemous headword with CompSci domain

If the headword is already part of the Slovene wordnet, no disambiguation is needed and the terms can be simply added as hyponyms to the existing Slovene hypernym. Also, in some cases one of the extracted multi-word terms was already in the Islovar dictionary. We can then use the English translation of the term to look up the correct hypernym and synset ID in PWN. Nevertheless there remain many cases where the correct sense must be picked manually (Table 7).

Term candidates	Slo. & Eng. hypernym possible synset IDs	Selected synset ID
nalaganje gonilnikov [loading drivers] nalaganje podatkov [loading data] nalaganje programov [software download] nalaganje strani [loading page]	nalaganje = loading ENG20-00671518-n [factotum] ENG20-13044298-n [transport]	to be selected manually

Table 7: Polysemous headword, ID to be selected manually

6 Discussion

Extracting terms from a large domain-specific Slovene corpus yielded the bulk of 74,310 term candidates. We keep only those that occur more than 5 times and where the headword and its English translation can be identified with reasonable accuracy. Some of these terms were already either in the Islovar dictionary or in SloWNet, however the large majority were new. Table 8 shows the number of terms successfully added to SloWNet.

Category	Number of terms
Already in SloWNet	29
Already in PWN	23
Already in Islovar	198
New	5150
Total	5400

Table 8: Total term candidates added to SloWNet

As has been described in the previous section, the tricky part is determining the correct sense of the potentially polysemous headword. While we use all the semantic information we can infer either from the domain label or the online dictionary, nearly half of all the headwords need to be disambiguated manually. In this respect our methodology could benefit significantly from additional context-based disambiguation procedures.

Category	Number of headwords
Monosemous	84
Headwords with CompSci domain	35
Headwords already in SloWNet	11
Headwords derived from MWE PWN	6
To be picked manually	136
Total	272

Table 9: Categories of headwords

7 Conclusions

We described an approach to improve the domain coverage of wordnet by enriching it with semi-automatically extracted multi-word terms. Our method was based on a combination of mono- and bilingual resources. A large monolingual domain-specific corpus was used as the source of terminology, and a smaller parallel corpus combined with a dictionary was used to provide translation equivalents of headwords. These are required in order to map the semantic structure of Princeton Wordnet onto the Slovene term candidates and thus integrate them into SloWNet.

Although the approach worked well and yielded many items of specialised vocabulary, manual validation would be needed in the headword disambiguation phase. It should be noted that an evaluation of monolingual term extraction lies beyond the scope of this paper and is not addressed, although the quality of the term candidates clearly influences the results of the experiment described.

In the future we intend to explore possibilities of automatically disambiguating polysemous headwords. An evaluation of the domain coverage of SloWNet will be performed within a Machine Translation application.

Bibliography

- [1] Bourigault, Didier (1993): Analyse syntaxique locale pour le repérage de termes complexes dans un texte. *Traitement Automatique des Langues*, vol. 34 (2), 105—117.
- [2] Diab, Mona & Philip Resnik (2002): An Unsupervised Method for Word Sense Tagging using Parallel Corpora. In *40th Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, July, 2002.
- [3] Dias, Gael & Nunes, S. (2004): Evaluation of Different Similarity Measures for the Extraction of Multiword Units in a Reinforcement Learning Environment. In M.T. Lino, M.F. Xavier, F. Pereira, R. Costa and R. Silva (eds): *Proceedings of the 4th International Conference On Languages Resources and Evaluation*, M.T. Lino, M.F. Xavier, F. Pereira, R. Costa and R. Silva (eds), Lisbon, Portugal, May 26-28. 1717—1721.
- [4] Dyvik, Helge (1998): Translations as semantic mirrors. In *Proceedings of Workshop W13: Multilinguality in the lexicon II*, pp. 24.44, Brighton, UK. The 13th biennial European Conference on Artificial Intelligence ECAI 98.
- [5] Erjavec, Tomaž & Darja Fišer (2006): Building Slovene Wordnet. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation LREC'06*. 24-26th May 2006, Genoa, Italy.
- [6] Erjavec, Tomaž & Ide, Nancy (1998): The MULTTEXT-East Corpus. In: *Proceedings of the First International Conference on Language Resources and Evaluation, LREC'98*. Granada, Spain.
- [7] Fellbaum, Christiane (1998): *WordNet: An Electronic Lexical Database*. MIT Press.
- [8] Fišer, Darja (2007): Leveraging Parallel Corpora and Existing Wordnets for Automatic Construction of the Slovene Wordnet. In: *Proceedings of the 3rd Language and Technology Conference L&TC'07*, 5-7 October 2007. Poznan, Poland.
- [9] Fišer, Darja (2008): Using Multilingual Resources for Building sloWNet Faster. In: *Proceedings of the 4th International WordNet Conference GWC'08*. 22-25th January 2008, Szeged, Hungary.
- [10] Ide, Nancy, Tomaž Erjavec & Dan Tufiş (2002): Sense Discrimination with Parallel Corpora. In *Proceedings of ACL'02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Philadelphia, 54-60.
- [11] Krstev, Cvetana, G. Pavlović-Lažetić, D. Vitas & I. Obradović (2004): Using textual resources in developing Serbian wordnet. In: *Romanian Journal of Information Science and Technology*. (Volume 7, No. 1-2), pp 147-161.
- [12] Piao, S., Rayson, P., Archer, D., Wilson, A. & McEnery, T. (2003): Extracting Multiword Expressions with a Semantic Tagger. In *Workshop on Multiword Expressions of the 41st ACL meeting*. 7-12 July. Sapporo. Japan. 49-57.
- [13] Steinberger Ralf, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş & Dániel Varga (2006): The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation*. Genoa, Italy, 24--26 May 2006.
- [14] Tiedemann, Jörg (2003): Recycling Translations - *Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing*, Doctoral Thesis. Studia Linguistica Upsaliensia I.
- [15] Tomokiyo, T. & Hurst, M. (2003): A Language Model Approach to Keyphrase Extraction. In *Workshop on Multiword Expressions of the 41st ACL meeting*. 7-12 July. Sapporo. Japan. 33-41.
- [16] Tufiş, Dan (2000): BalkaNet - Design and Development of a Multilingual Balkan WordNet. In: *Romanian Journal of Information Science and Technology Special Issue* (Volume 7, No. 1-2).
- [17] Vintar, Špela (2004). Comparative Evaluation of C-value in the Treatment of Nested Terms. *Memura 2004 - Methodologies and Evaluation of Multiword Units in Real-World Applications* (LREC 2004), pp. 54-57.
- [18] Vintar, Špela; Fišer, Darja (2008) Harvesting Multi-Word Expressions from Parallel Corpora. In: *Proceedings of the 6th International Conference on Language Resources and Evaluation LREC'08*. 28-30th May 2008, Marrakech, Morocco.

The Digitisation and Deployment of the Slovenian Biographical Lexicon

Jan Jona Javoršek¹, Tomaž Erjavec², and Petra Vide Ogrin³

¹ Department of Experimental Particle Physics
Jožef Stefan Institute,
Jamova cesta 39, Ljubljana, Slovenia
jan.javorsek@ijs.si

² Department of Knowledge Technologies,
Jožef Stefan Institute
tomaz.erjavec@ijs.si

³ Slovenian Academy of Sciences and Arts, Library
Novi trg 3, Ljubljana, Slovenia
petra.vide@zrc-sazu.si

Abstract. We present the digitisation and deployment of the Slovenian Biographical Lexicon (SBL), an extensive publication and an important resource for encyclopaedic and reference editions and research in the Slovenian humanities, social sciences and history of the natural sciences. The paper presents the methodology, based on open standards and software, that we used produce a freely available on-line digital re-edition of SBL. In the process of digitalization, manually corrected OCR has been semi-automatically converted to an XML-based Text Encoding Initiative encoding (TEI P5). Its extensive annotation vocabulary, notably from the biographical and prosopographical modules, has been used to semi-automatically mark-up as much data as possible. The resulting XML document has become the data resource of an online digital repository based on Fedora Commons platform, where we implemented an infrastructure of XML processing methods and a Lucene/SOLR based search engine to produce a full-fledged web application and search engine with browser, metadata and web application interfaces.

1 Introduction

The Slovenian Biographical Lexicon (SBL, [1]) was conceived as a publication that was to give an accurate picture of Slovenia's cultural life, from its beginnings up to the contemporary time by including everybody who participated in the cultural development, either of Slovenian origin, born on Slovenian soil or influencing Slovenian cultural life.

This broad aim resulted in a list of 2,335 names, mostly from the fields of humanities and social sciences, proposed by the original editorial board. In its long history, this list has been changed and expanded: especially after WW2 the focus was shifted to reflect the “increasing development of natural sciences, modern technologies and their applications, as part of the spiritual superstructure” (SBL, vol. 15, 1991). In spite of several eliminations from the original list, with the publication of the final volume in 1991, SBL comprises as many as 5,036 biographical entries, with more than 5,100 persons covered. Since the publication was published sequentially over almost 70 years, it is important to note that the criteria for different published volumes have varied significantly.

The aim of the publication was to be both informative and exhaustive; therefore much substantial information had to be included in rather short articles (with several longer exceptions). The data in the articles was checked against relevant historical materials and pre-existing publications., e.g. biographical and other dates are always compared to dates in registers and other primary documents, literary citations are compared with originals, sources are cited at the end of the articles and the publication includes an index of all person names that appear in the articles and a list of abbreviations.

As a result, SBL contains a surprising amount of high-quality information and references and remains to this day a precious resource for encyclopaedic and reference editions and research in the Slovenian humanities, social sciences and history of the natural sciences. It has, however, two severe drawbacks: the original edition has quickly become difficult to find and the information, once published, has never been updated.

In Figure 1 we give an excerpt from the start of the printed edition of the SBL, where we can see the type and amount of biographical facts given for a person, as well as the rather heavy use of abbreviations, which become even more frequent in the later volumes.

A

Abraham, škof v Freisingu na Bavarskem, izvoljen po smrti škofa Lamberta (u. 19. sept. 957), posvečen 21. dec. 957, u. 26. maja 994. V začetku svojega škofovanja je bil pristaš cesarja Otona I. in bavarske vojvodinje Judite ter njenega sina vojvode Henrika II., cesarjevega nečaka. Po smrti Otona I. je izpremenil stališče in se pridružil bavarskemu vojvodu Henriku II., kateri je stremel po osamosvojitvi svoje obširne vojvodine od cesarjeve oblasti, skušal pritegniti kolonizacijsko ozemlje ob Donavi in med alpskimi Slovenci pod svojo interesno sfero ter ustvariti tesne zveze z Italijo, kjer je bila Bavarski pridružena Veronska marka. Upor bavarskega vojvode proti cesarju se je poleti 974 izjalovil, A. je bil za kazen prejkone avg. 974 pregnan v Corvey na Westfalskem, a se je kesneje zopet pomiril s cesarjem. — Pod A. je dobila freisinška cerkev obširen zemljiški kompleks v Kranjski marki ok. današnje Skofje Loke ob porečju selške in poljanske Sore (prva darovnica ces. Otona II. 30. jun. 973, razširjena 23. nov.

koroško, oz. celo slovensko pokolenje A. in za njegovo bivanje na koroško-slovenskih tleh nimamo nobenih verodostojnih dokazov. O A. pokolenju piše kot prvi šele nekritični koroški zgodovinar Jakob Unrest (u. 1500), o njegovem koroškem pregnanstvu goriški historiograf Martin Bauzer (u. 1668); vesti obeh slone na kesnih in lokalnih tradicijah. Reči se da le toliko, da so slov. teksti mogoče nastali še za vladikovanja A., paleografska analiza pisav slov. tekstov kaže na nastanek nekako v razdobju 975—1025. — Prim.: C. Meichelbeck, *Historiae Frisingensis* tomus I, Augustae Vindelicorum et Graecii 1724, 173—189; B. Kopitar, *Glagolita Clozianus*, Vindobonae 1836, XXXIV, XLI, XLII; Hundt v *Abh. der k. bayer. Akad. der Wiss.*, III. Cl., 14, 2. Abth.; R. Nahtigal, *ČJKZ*, I (1918); M. Kos, *istotam*, IV (1924).

M. Kos.

Abram Filip, sodnik in sodni upravnik, r. 1835 v Štanjelu pri Sežani, u. 1. apr. 1903 na Dunaju. Gimnazijo je obiskoval v Gorici in Benetkah (stric mu je bil dvorni

Figure 1: Excerpt from the printed edition of SBL.

The present project of digitalization of SBL has been started by the Slovenian Academy of Sciences and Arts (SASA) and the Scientific Research Centre of the SASA to make this important resource available again, this time in the form of a freely accessible on-line edition, and has been based on previous similar projects undertaken in cooperation with the Jožef Stefan Institute [2]. This paper describes the steps taken on the path from the original publication towards a fully searchable and cross-indexed on-line edition. The first steps of the process, from digitalization using OCR and manual revision to semi-automatic encoding and mark-up in the form of Text Encoding Initiative XML document (TEI P5), will be summarized¹ before we consider the methodology and implementation of the on-line digital repository for the digital edition that can function as a flexible web application. We will present several possibilities offered by our implementation, some of which remain for further experiments. It is worth noting at this point that the digital edition in its current form is available for testing at the URL <http://nl.ijs.si/fedora/sbl/>.

2 Encoding the SBL

The encoding of SBL is based on open standards, in particular the Text Encoding Initiative Guidelines. TEI produced recommendations or guidelines for the creation and processing of electronic texts for better interchange and integration of scholarly textual data in all languages and from all periods [4]. We used the latest edition of TEI Guidelines, TEI P5, published in 2007 [5], since it provides important new encoding features, including new support for manuscript descriptions, multimedia and graphics, stand-off annotation, representation of data pertaining to people and places and improved specifications for encoding textual alternatives.

TEI P5 also takes advantage of the power of XML schema languages, so that other XML tag-sets, such as MathML or SVG, can now be referenced from within a TEI document and a TEI document can be embedded within other types of XML documents, such as METS and MODS records. This turned out to

¹ See [3] for a more detailed treatment of this topic.

be crucial for the implementation of our on-line repository, since this makes TEI a well-behaved XML citizen, able to take part in any, however complex, XML processing chains and in composite documents.

3 Up-translation and structure of an SBL article

The vast majority of SBL articles present information on the life and actions of a single person, while some describe well known families, detailing life and work of several members of the family. An article usually starts with the name of the person or the family, its variants, mostly those used towards the end of their life or the most generally used, followed by a chronological summary of their life and activity, including birth, death, locations, occupations, activities etc. An article consists of one (usually) or many paragraphs, depending on the exhaustiveness of the article, and is written in dense language, using abbreviations wherever possible, ending with a brief bibliography and other materials relevant to the person, such as portraits or photographs.

The text of the articles has been digitized and manually revised to fix OCR errors before it was automatically converted into a basic TEI-XML format. In the next stage, those segments of the text that needed to be marked up but could not be identified automatically were tagged manually, in particular with details such as different variants of names (linguistic and orthographic variations, married names, ecclesiastic names and titles, pseudonyms, complex name parts in the case of foreign names and names with denotation of nobility etc.), making the process slow and error-prone. Since the original data was not normalized, considerable effort had to be spent to achieve high quality TEI XML mark up, and some work with data normalization is still ongoing. The major aspects of this conversion process have been reported in more detail in [3]. In this manner, essential information about the subject of the article and its bibliographical section have been encoded within special purpose elements from TEI P5 biographical and prosopographical modules.

```

<div>
  <listPerson>
    <person n="main">
      <sex value="1"/>
      <persName>
        <name>Abraham</name>
        <roleName type="eccl">škof</roleName>
      </persName>
      <occupation>duhovnik</occupation>
      <death><date when="0994-05-26">26. maja 994</date>
      </death>
    </person>
    <person n="author">
      <sex value="1"/>
      <persName key="M. Kos.">
        <surname>Kos</surname>
        <forename>Milko</forename>
      </persName>
    </person>
  </listPerson>
  <p>Abraham, škof v Freisingu na Bavarskem, izvoljen po smrti škofa
  Lamberta (u. 19. sept. 957), posvečen 21

```

Figure 2: An SBL article in TEI P5 encoding.

As can be seen in Figure 2, each article is represented as a <div> containing a <listPerson> element and one or more paragraphs (<p>). The paragraphs contain the text of the printed SBL, while <listPerson> contains the semi-automatically extracted information from the article text. The most important element is <person type="main">, which details facts about the subject of the article, such as names, sex, birth/death date and location, locations of activities, occupations or activities etc. <listPerson> also contains meta data, such as the author of the article, revision status etc. While not shown in Figure 2, the paragraph elements are followed by <listBibl> element with the extracted bibliographical data of the article.

Obviously, the actual structure to a certain extent depends on the information of that particular article, and so the type and number and elements varies considerably (i.e. marriage, ordination, exile, further education, number of occupations, residence, active period etc.). This makes any mapping into a more formal structure, i.e. a relational database, at least awkward.

There is a number of further details that could be extracted from the text but meticulous manual intervention would be required to achieve suitable accuracy. The most important of these are activities undergone by the person, encodable in the <occupation> tag or tags, and locations and times of these activities, encodable by the <floruit> tag.

4 Anatomy of an XML-based Document Repository

We have evaluated a number of possible platforms to serve as the base of an on-line web edition of SBL with integrated query and search tools. One possibility considered was PhiloLogic², a full-text search, analysis, and retrieval tool developed by the ARTFL Project³ and the Digital Library Development Center at the University of Chicago. PhiloLogic uses an abstract representation of document structure, projecting the XML data into sets of related database tables, so that the application can search document structure and refine word searching by using the XML structure [6]. However, in the end, mainly due to its greater flexibility and possibility of integration with other software, we have chosen the Fedora Commons platform [7], an extensible framework for storage, management and dissemination of complex objects and object relationships implemented as a portable Java web application [8]. Fedora Commons became the repository on top of which we created a digital library of biographical articles with browsing, searching and querying interfaces: a digital library that is presented as an on-line web service and application.

Fedora Commons represents its digital objects as a collection of data streams, where each document is specified as an XML document (Fedora Commons has a native format, called FOXML, but also supports the Metadata Encoding Transmission Standard METS⁴). Data streams can be of different types: formally, they can be created as embedded XML documents, as managed independent files in the repository (used for binary files, e.g. images or PDF documents) or as external URI-specified documents. In addition, each object has a number of infrastructure-supporting data streams, all in the form of embedded XML documents. Among them, there is a Dublin Core⁵ data stream to contain object meta-data, an RDF⁶ data stream to declare inter-object relationships, and internal FOXML revision specifications to allow tracking of object history.

Fedora Commons objects have dissemination methods (analogues to object or class methods in object-oriented systems), implemented as web application interfaces to objects and their contents (both REST and SOAP interfaces are supported). Since version 3 of the platform, a new Content Model Architecture has been introduced under which dissemination methods are specified with three special objects types: Content Model objects specify available methods and necessary data streams for the dissemination methods they declare, Service Definition objects define a web API (Application programming interface) for dissemination methods and Service Deployment objects use WSDL⁷ (Web Services Description Language) to specify the actual web application API calls necessary to execute a dissemination method request (cf. Fedora Commons Content Model Architecture documentation for version 3⁸).

All the required information, such as the necessary data for each dissemination method call, supported data-types and the manner of invocation to produce the result, are specified in the form of embedded XML documents in the three types of objects, which are otherwise structured as any other object in the repository. To add dissemination methods from one or several different Content Models, it is therefore

² <http://philologic.uchicago.edu/>

³ <http://humanities.uchicago.edu/orgs/ARTFL/>

⁴ <http://www.loc.gov/standards/mets/>

⁵ <http://dublincore.org/>

⁶ <http://www.w3.org/RDF/>

⁷ <http://www.w3.org/TR/wSDL/>

⁸ <http://fedora-commons.org/confluence/display/FCR30/Content+Model+Architecture>

sufficient to add a special relationship to an object, referring to the Content Model in question. Its dissemination methods will become available under the access URI of the object, contained in a path element of the Content Model's name. Furthermore, even the core Fedora Commons features, such as object introspection and direct data stream access, are implemented in this way using the default Content Model.

This extensible and standards-based Content Model Architecture in combination with a number of web services, namely the SAXON XSLT processor,⁹ an image manipulation library, an RDF query interface to the object relationship RDF store, a simple search interface to Dublin Core meta-data and object properties and the Fedora Generic Search interface to a number of optional search engines, provide an infrastructure for development of rich application interfaces and complex multi-layered digital repositories using standard technologies and XML workflows.

5 From XML Datasources and Workflows to an Online Application

In the Fedora Commons framework, each dissemination method is realized as a web application call (using REST or SOAP methods) with a number of arguments, usually one or several of the object's data streams. A data stream can also be specified as an URI, referring to another dissemination method and thus resulting in a chain of processing calls. While this approach can be used with binary data, i.e. to apply a number of transformations to an image, it is usually used to create an XML workflow. Such XML workflows have become the backbone of our application since they allow us to pool together XML data from several sources, such as object data streams and object relationship query results, to form the final XML response, usually in the form of an XHTML rendering in the user's browser.

Essentially, there are two kinds of data objects in our application: collections and articles. Collections use inter-object relationships to represent different views of the data to the user: these are the top-level objects, containing all the other views, letter-objects and volume-objects that allow browsing alphabetically through lists of articles or browsing through the articles in the units of their publications, and search-result objects that take a search query and represent its results as a collection of objects.

SBL article objects are much simpler – in essence they transform the TEI data to an XHTML-encoded web page. The resulting page (an example is given in Figure 3) also contains a number of context-dependent links, including facilities such as a search interface, browsing links (previous, next), and possibly links to instant search queries that provide article lists representing e.g. members of the same occupation class, members of the same generation or all the contemporaries of the subject of the article. There are also other links, such as a link to all the articles by the same author, accessed via the author name, etc.

All the article objects also provide direct links to their TEI XML source and their meta data in Dublin Core encoding [9,10], transformed from the structures in the <listPerson> element of each article. This seemingly trivial feature is actually essential: it means that it is possible, through a public API, to access all of the original TEI document data and all of the meta data, structured in a standard-compliant way. In combination with the platform-integrated support for the Open Archives Initiative Metadata Harvesting Protocol (OAI-MHP¹⁰), this makes our digital repository very easy to integrate with other systems [11,12].

There is a number of simpler objects, mostly renderings of parts of TEI header information, that simply convey meta data about the whole collection in an accessible form – but they all obey the same logic and use the same infrastructure.

This architecture, in spite of its relative simplicity, has allowed us to construct a flexible and efficient user interface. In general, all the context information has become clickable or otherwise accessible through simple links, making the browsing interface extremely powerful. But the true power of the implementation comes from its search interface.

⁹ <http://saxon.sourceforge.net/>

¹⁰ <http://www.openarchives.org/OAI/openarchivesprotocol.html>



Figure 3: The XHTML rendering of an SBL article with links to table of contents, Dublin Core data and TEI source and the simple search mask.

6 Search and Query Interface

Fedora Commons provides an integrated search system, capable of simple searches on Dublin Core metadata and object properties, but a much more powerful system, Fedora Generic Search, is also available; its power derives from the fact that it provides native Fedora Commons interfaces to external search systems.

We have implemented Fedora Generic Search on top of SOLR,¹¹ a search system based on Apache Lucene¹² search and indexing library. In this set-up, Lucene library can use Fedora Commons API to index the document contents, using specially crafted rules (in the form of XSLT stylesheets) to break up the documents in a number of searchable text fields, and the repository gains a search interface with the full power of the Lucene query language [13], while SOLR takes over the interface and formatting of query results as an easy-to parse XML list.

The power of the interface has been most useful while crafting special queries, such as the context links for different SBL article features, but due to the complexity involved in the use of the flexible Lucene query language, this is hardly the optimal solution for the average user of the system.

To solve this problem we have implemented two user search interfaces: the simple search is targeted at the most often requested fields, namely the subjects' names, places of their birth and death, and their occupations. This interface also accepts the full Lucene syntax, so it can be used both for simple searches by general users and for possibly very complex queries by advanced users.

The secondary, advanced interface, shown in Figure 4, can be accessed by a click on an expansion button. It presents a form with several fields that enable an average user to easily compose fairly complex queries, selecting different indexed fields and even using advanced features such as full-text searches, proximity searches, number or date ranges, fuzzy searches etc. This interface is implemented by a secondary script that parses the form and converts its data into a single Lucene query string. Our initial testing with users has been reasonably successful: it makes it trivial to find, for example, female writers who lived in the period between 1830-1860 in Ljubljana, or philosophers born in Maribor.

The same system can be used for experimental research on the data, especially if one wants to analyse the particulars of the original SBL publication. It is now easy to compare, for example, the average length of articles with the number of articles contributed by an author (showing the difference between regular

¹¹ <http://lucene.apache.org/solr/>

¹² <http://lucene.apache.org/>

contributors and specialists for narrow fields), to plot the average length of life by year of birth or by occupation (and find extreme cases, such as exceptionally short life spans for heroes of WW2 and the revolution). In fact, a number of such queries with graphic interfaces, together with usual on-line features, for example a listing births and deaths on the current date, are being included in the current version of the application. Obviously, this opens up possibilities for further research that falls outside the scope of the present paper.

Figure 4: The advanced search interface for SBL with fields for: person, sex, occupation, dates and places, author, full text search; exact/fuzzy searches, logical connective between search fields. Dates and places are of birth, death, and period of activity. Note also links to the Foreword, TEI header; random person, list of families, and born/died on the date of the search.

7 Conclusions

The paper presented a specific application of the general methodology for building digital text repositories on the basis of open encoding and software standards. The foundation on which the SBL rests is XML and TEI P5 Guidelines, which is open and maintained under an international consortium. The mode of access and presentation, on the other hand, rest on open standards and software mostly developed under W3C and the Apache project. This methodology makes for durable digital resources, quick development time for Web deployment, and simple integration with other XML and Internet based protocols.

The project of the digital edition of SBL has reached production stage, where a complete digital edition is hosted in an on-line digital library and an on-line user interface is available. The specific goal of the project has been achieved: the valuable reference information captured in the lexicon is now again available to the research community and general public, this time enhanced with cross-linking, context information and an advanced query and retrieval system that facilitates its use.

But there are several objectives that are still being worked on. Primarily, we would like to extract and encode further extents of information, since we now have a functional framework that enables us to use the information, once marked up, to provide further features. The foremost points are the two crucial pieces of information: the subject's name and occupation, representing the two identifications used to most often find subjects of interest. The work of manual annotation of name parts with further corrections in spelling and markup of names is being finished at the time of this writing. At the same time, we are working on normalization of occupation specifications (there are more than 1500 different strings for occupations or activities in the SBL articles) and introducing a simple taxonomy to enable meaningful grouping and searching. Furthermore, we have plans to normalize names of places and annotate them with geographic identifiers. Since we want to mark-up any names mentioned in the article texts, we are planning to develop a Named Entity Recognition (NER) tool, [14,15] for Slovenian, and we have gathered substantial databases of names and places for this purpose.

Another issue is the heavy use the SBL makes of abbreviations. While this was a sensible strategy for the printed edition, it only makes the on-line version harder to read. We would therefore like to expand the abbreviations; however, this is a non-trivial process, as the abbreviations need to be disambiguated and expanded into their properly inflected forms, which are, of course, context dependent.

In closing, we are happy to report that our system will be used by ongoing and new projects in the field of biographic publications in Slovenia and will likely become the platform for the digital publication of the Slovenian Biographical Lexicon 2 and for the Slovenian Biographical Hub which is to integrate most available resources in this domain.

References

- [1] Cankar, Izidor et al. (eds). Slovenski biografski leksikon. Ljubljana: Slovenska akademija znanosti in umetnosti, 1925-1991
- [2] Erjavec, Tomaž; Ogrin, Matija. Digital Critical Editions of Slovenian Literature: an Application of Collaborative Work Using Open Standards. *From Author to Reader: Challenges for the Digital Content Chain: proceedings of the 9th ICCA International Conference on Electronic Publishing*, Arenberg Castle / Dobrova, M.; Engelen, J. (eds.). Leuven: Peeters, 2005, 151-156
- [3] Vide Ogrin, Petra; Erjavec, Tomaž: Towards a Digital Edition of the Slovenian Biographical Lexicon. In: *The Future of Information Sciences: Digital Information and Heritage : Proceedings of the 1st International Conference The Future of Information Sciences - INFUTURE 2007*. Seljan, Sanja; Stančić, Hrvoje (eds.). Zagreb: Filozofski fakultet, Sveučilište u Zagrebu, 2007. 115-124.
- [4] Burnard, Lou. Encoding standards for the electronic edition. *Znanstvene izdaje in elektronski medij: razprave*. Ogrin, M. (ed.). Ljubljana: Založba ZRC, ZRC SAZU, 2005, 25-42
- [5] Burnard, Lou; Bauman, Syd. TEI P5: Guidelines for Electronic Text Encoding and Interchange (TEI P5). Text Encoding Initiative Consortium. HTML Version. Oxford, 2007. <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/> (2009-08-19)
- [6] Cooney, Charles M. et al.. Extending PhiloLogic. April 2007 <http://www.digitalhumanities.org/dh2007/abstracts/xhtml.xq?id=175> (2007-09-09)
- [7] The Fedora Project: An Open-Source Digital Repository Management System, <http://fedora-commons.org/>
- [8] Lagoze, Carl; Payette, Sandy; Shin, Edwin; Wilper, Chris. Fedora: an architecture for complex objects and their relationships, *International Journal on Digital Libraries*, 6/2, 2006, 124-138.
- [9] Miller, Eric, Brickley, Dan, 2001: Expressing Simple Dublin Core in RDF/XML, Dublin Core Metadata Initiative Proposed Recommendation.
- [10] Liddy, Elizabeth D., Allen, Eileen, Harwell, Sarah, Corieri, Susan, Yilmazel, Ozgur, Ozgencil, Ercan N., Diekema, Anne, Mccracken, Nancy, Silverstein, Joanne in Sutton, Joanne, 2002: Automatic metadata generation & evaluation. *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM Press, 401-402.
- [11] Benjamin, Wolfgang Nejd, Siberski, Wolf , 2002: OAI-P2P: A Peer-to-Peer Network for Open Archives. *Workshop on Distributed Computing Architectures for Digital Libraries - ICPP2002*.
- [12] Ward, Jewel, 2004: Unqualified Dublin Core usage in OAI-PMH data providers. *OCLC Systems & Services*, 20(1), 40-47.
- [13] Hatcher, Erik, Gospodnetic, Otis 2004: *Lucene in Action*. New York: Manning Publications.
- [14] Jackson, Peter; Moulinier, Isabelle. *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*. Amsterdam: John Benjamins, 2002, 180-185
- [15] Bekavac, Božo. Strojno obilježavanje hrvatskih tekstova – stanje i perspektive. *Suvremena lingvistika*. 53-54 (2002), 173-182. (2007-09-14)

Bulgarian-Polish-Lithuanian Corpus –Problems of Development and Annotation¹

Ludmila Dimitrova¹, Violetta Koseska², Danuta Roszko², Roman Roszko²

¹ Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia

² Institute of Slavic Studies, Polish Academy of Sciences, Warsaw

Abstract. The paper shortly describes the first Bulgarian–Polish–Lithuanian (for short, BG–PL–LT) experimental corpus, currently under development only for research. The trilingual corpus comprises two corpora: parallel and comparable. We focused our attention on the morphosyntactic annotation of the parallel trilingual corpus, according to the Corpus Encoding Standard (CES). We briefly discuss the tagsets for corpora annotation from the point of view of possible unification in the future. Next, we review the Part-of-Speech (POS) classification of the *participle* in the three languages, in comparison to another POS, the *adjective*. Some examples are presented.

1 Introduction

One of the main problems in human communication is the presence of a huge variety of written and spoken languages in the world. Finding ways to support the connection of people from different ethnical parts of the world is becoming more and more important. Due to the recent development of information and communication technologies and the increased mobility of people around the globe, the number of bilingual electronic dictionaries, in which one of the languages is English, has increased extraordinarily. One cannot expect however that all people know English to communicate with each other, especially if their native languages (Bulgarian and Polish) belong to the same language family. An Internet search shows that no electronic dictionaries exist at all for pairs of languages such as Bulgarian-Polish or Bulgarian-Lithuanian. Traditional printed paper dictionaries are either an antiquarian rarity (the most recent Bulgarian-Polish and Polish-Bulgarian dictionaries were published more than 20 years ago) or have never been published at all (Bulgarian-Lithuanian). For the creation of a bilingual electronic or online dictionary for Bulgarian, Polish and Lithuanian an electronic corpus is necessary which will provide the material for lexical database, supporting the dictionary and its subsequent expansion and update.

On the one hand, it is interesting to note that two Slavic languages are compared to a Baltic language (Lithuanian). Comparative and contrastive studies of Polish and Bulgarian as well as Polish and Lithuanian have been already conducted, but up to the best of our knowledge no such studies exist for Bulgarian and Lithuanian. On the other hand, the three languages are marginally present in the EU because of the later ascension of the three countries to the EU. Thus we expect a new and interesting scientific problem in front of us and hope that our studies will find a wider application.

2 From Bilingual to Trilingual corpus

In recent decades many multilingual corpora were created in the field of corpus linguistics, such as the MULTEXT corpus [7], the MULTEXT-East corpus, annotated parallel and comparable, (MTE for short), an extension of the corpus MULTEXT with six Central and Eastern European (CEE) languages [2], ParaSol, a parallel and aligned corpus of Slavic and other languages (so-called Regensburg Parallel Corpus) [23], Italian-German parallel corpus, a collection of legal and administrative documents written in Italian and German, due to the equal status of the both languages in South Tyrol [9], Hong Kong bilingual parallel English-Chinese corpus of legal and documentary texts [6], etc.

The MTE project has developed a multilingual corpus, in which three languages: Bulgarian, Czech and Slovene, belong to the Slavic group. The MTE model is being used in the design of the first Bulgarian-Polish corpus [4], [5], currently under development in the framework of the joint research project “Semantics and Contrastive linguistics with a focus on a bilingual electronic dictionary” between Institute

¹ The first Bulgarian-Polish corpus, currently under development in the framework of the joint research project “Semantics and Contrastive linguistics with a focus on a bilingual electronic dictionary” between IMI-BAS and ISS-PAS. The study and preparation of this paper have been partly supported by the EC’s Seventh Framework Programme [FP7/2007-2013] under the grant agreement 211938 MONDILEX.

of Mathematics and Informatics—Bulgarian Academy of Sciences and Institute of Slavic Studies—Polish Academy of Sciences, coordinated by L. Dimitrova and V. Koseska. This bilingual corpus supports the lexical database (LDB) of the first experimental online Bulgarian-Polish dictionary [3].

2.1 Bulgarian-Polish corpus

The Bulgarian–Polish corpus consists of two parts: a parallel and a comparable corpus. All collected texts in the corpus are texts published in and distributed over the Internet. Some texts in the ongoing version of the corpus are annotated at paragraph level.

The **Bulgarian–Polish parallel corpus** includes two parallel sub-corpora:

1) a *pure* Bulgarian–Polish corpus consists of original texts in Polish – literary works by Polish writers and their translation in Bulgarian, and original texts in Bulgarian - short stories by Bulgarian writers and their translation in Polish.

2) a *translated* Bulgarian–Polish corpus consists of texts in Bulgarian and in Polish of brochures of the EC, documents of the EU and the EU-Parliament, published in Internet; Bulgarian and Polish translations of literary works in third language (mainly English).

The **Bulgarian–Polish comparable corpus** includes texts in Bulgarian and Polish: excerpts from newspapers and textual documents, shown in internet, excerpts from several original fiction, novels or short stories, with the text sizes being comparable across the two languages. Some of the Bulgarian texts are annotated at “paragraph” and “sentence” levels, according to CES [8].

2.2 Bulgarian–Polish–Lithuanian corpus

The first Bulgarian–Polish–Lithuanian (for short, BG–PL–LT) corpus (currently under development only for research) contains more than 3 million words and comprises two corpora: parallel and comparable.

The **BG–PL–LT parallel corpus** contains more than 1 million words. A part of the parallel corpus comprises original texts in one of the three languages with translations in two others, and texts of official documents of the European Union available through the Internet. The main part of the parallel corpus comprises texts (fiction, novels, short stories) in other languages translated into Bulgarian, Polish, and Lithuanian. When we have provided the electronic text of the original literary work or its translation, we include it as well in the corpus.

It turned out that it is extremely difficult to find electronic texts of translations from Bulgarian to Lithuanian or *vice versa* – the two languages are spoken by small nations in comparison to other languages of the EU and are distributed in remote areas of Europe. It can be assumed (provisionally of course) that the Polish language ‘builds a bridge’ between them: for the pairs of languages Bulgarian-Polish and Polish-Lithuanian one can find freely available translations on the Internet.

We plan to annotate the BG-PL-LT parallel corpus according to the standards for morphosyntactic annotation of digital language resources.

The **comparable BG–PL–LT corpus** includes: (1) texts in Bulgarian, Polish and Lithuanian with the text sizes being comparable across the three languages, mainly fiction, and (2) excerpts from electronic newspapers, distributed via Internet and with the same thematic content.

The main goal in collecting the trilingual corpus is the design and development of a BG–LT digital dictionary based on the BG-PL digital online dictionary. The corpus will provide a sample of the vocabulary, which is to be included in an initial experimental versions of BG–LT digital dictionary.

The structure of the parallel corpus groups texts according to content. Every group contains three parts (respectively four if the original language is different from the languages in the corpus). A detailed description of the corpus is provided for clarification to the user.

An excerpt of the description of the trilingual parallel corpus follows:

BG Bulgarian: Станислав Лем, *Соларис*. Translated by Андреана Радева. Отечество, София, 1980.

PL Polish: Stanislaw Lem, *Solaris*. Wydawnictwo Literackie, Kraków, 1961.

LT Lithuanian: Stanislavas Lemas, *Soliaris*. Translated by Giedrė Juodvalkytė. Vaga, Vilnius, 1978.

// EN: *Stanislaw Lem, Solaris* //

Some of the texts have been annotated at paragraph level. This allows texts in all three languages and in pairs (BG–PL, PL–LT, BG–LT, and *vice versa*) to be aligned at paragraph level in order to produce aligned three- and bi-lingual corpora. “Alignment” means the process of relating pairs of words, phrases, sentences or paragraphs in texts in different languages which are translation equivalent. One may say that “alignment” is a type of annotation performed over parallel corpora.

Excerpts of texts of the 3-languages parallel corpus, marked at paragraph level follow:

Bulgarian:

<p>Вместо отговор Гандалф гръмогласно подвикна на коня си:</p>

<p>- Напред, Сенкогрив! Трябва да бързаме. Няма време. Виж! Сигналните кладии на Гондор горят, зоват за помощ. Войната е избухнала. Виж, огън бушува над Амон Дин, пламък покрива Ейленах, сигналът бърза на запад: Нардол, Ерелас, Мин-Римон, Каленхад и Халифириен на роханската граница.</p>

Polish:

<p>Zamiast odpowiedzię hobbitowi, Gandalf krzyknęł głośnie do swego wierzchowca:</p>

<p>- Naprzód, Gryfie! Trzeba się spieszyć. Czas nagli. Patrz! W Gondorze zapalono wojenne sygnały, wzywają pomocy. Wojna już wybuchła. Patrz, płoną ogniska na Amon Din, na Eilenach, zapalają się coraz dalej na zachodzie! Rozbłyska Nardol, Erelas, Min-Rimmon, Kalenhad, a także Halifirien na granicy Rohanu.</p>

Lithuanian:

<p>Užuot atsakęs Gendalfas garsiai riktelėjo žirgui:</p>

<p>- Pirmyn, Žvaigždiki! Reikia skubėti. Laiko nebeliko. Žiūrėk! Jau dega Gondoro laužai, prašo pagalbos. Karo kibirkštis įžiebta. Matai, ant Amon Dino dega ugnis, liepsnoja ir Eilenachas, dar toliau vakaruose - Nardolas, Erelasas, Minas Rimonas, Kalenhadas ir Halifirienas prie Rohano sienos.</p>

//EN: For answer Gandalf cried aloud to his horse. ‘On, Shadowfax! We must hasten. Time is short. See! The beacons of Gondor are alight, calling for aid. War is kindled. See, there is the fire on Amon Dîn, and flame on Eilenach; and there they go speeding west: Nardol, Erelas, Min-Rimmon, Calenhad, and the Halifirien on the borders of Rohan. (Part 3, Book 5 of *The Return of the King* of Tolkien’s *The Lord of the Rings*)//

3 Corpus annotation and problems related to POS classification

Corpus annotation is the process of adding linguistic information in an electronic form to a text corpus [8], [10]. We would like to mention the following two most common types of corpus annotation: **morphosyntactic annotation** (also called *grammatical tagging* or *part of speech (POS) tagging*) and **lemma annotation** (where each word in the text is associated with the corresponding lemma). Lemma annotation is closely related to morphosyntactic annotation. Morphosyntactic annotation (POS tagging, where each word in the text is associated with its grammatical classification) is the task of labeling each word in a sequence of words with its appropriate part-of-speech. Words are often ambiguous with respect to their POS. For example, in Bulgarian the neuter singular forms of most adjectives serve double duty as

adverbs:

BG: гръмогласно //EN: loud(-voiced), uproariously, voice of thunder //:

(1) *гръмогласно* //loud(-voiced)//→ POS specifications: adjective, Gender: neuter, Number: singular, Definiteness: no.

MTE MorphoSyntactic Descriptor (MSD) for this adjective is A--ns-n.

(2) *гръмогласно* // uproariously, voice of thunder, cried aloud // → POS: adverb, Type: adjectival.

MTE MSD for this adverb is Ra.

The set of POS tags is called tagset. The size and choice of the tagsets vary across languages. The classical POS tagging system is based on a set of parts of speech including noun, adjective, numeral, pronoun, verb, participle, adverb, preposition, conjunction, interjection, particle, and often (depending on the language) article, etc. Of course, morphologically rich languages need more detailed tagsets that reflect to various inflectional categories. The POS classification varies across different languages. Often there is more than one possible POS classification for a given language.

The applications of the morphosyntactic annotation include lexicography, parsing, language models in speech recognition, disambiguation clues for ambiguous words (machine translation), information retrieval, spelling correction, etc.

Here we would like to show that one cannot formally go about a direct use of the morphosyntactic annotation of a multilingual corpus. An in-depth contrastive study of specific phenomena in the respective languages is necessary. Next we attempt to perform a comparison of the morphosyntactic characteristics of the words of parallel texts across the three languages from the point of view of a possible future unification. We will briefly review the POS classification of the *participle* (one of the important verbal forms) in the three languages, in comparison to another POS, the *adjective*.

3.1 Functions of the participle

The classification of a participle, not only as a verb form, is an important problem: the role of the participle varies significantly across languages, because its properties and functions are different. In contrast to English, for instance, where the participle are invariant, in the Slavic languages the forms of the participles are inflected and contain information about the aspect and tense of the verbal form. As is well-known the information about the aspect is important for the Slavic languages, but does not exist in English. Bulgarian, Polish and Lithuanian distinguish between the following functions of the *participle* form: predicative function, attributive function and semi-predicative function or adverbial function, which are illustrated by the following examples:

(1) Examples of predicative function of the participle

BG: украсен // PL: ozdobiony // LT: papuošta [neuter], papuoštas [masculine] //EN: *decorated*//:

BG: Коридорът е хубаво **украсен**.

PL: Korytarz jest ładnie **ozdobiony**.

LT: Koridorius gerai **papuošta**. / Koridorius gerai **papuoštas**.

(EN: *The corridor is beautifully decorated.*)

(2) Examples of attributive function of the participle:

BG: пишещ // PL: piszący // LT: rašantis // EN: *one who wrote* //, in the sentences:

BG: **Пишещият** тези писма **старец** е осемдесетгодишен.

PL: **Piszący** te listy **starzec** jest osiemdziesięcioletkiem.

LT: **Rašančiam** tuos laiškus **seneliui** aštuoniasdešimt metų.

(EN: *The old man who wrote these letters is eighty years old.*)

(3) Examples of the semi-predicative function:

BG: пишейки // PL: pisząc // LT: rašydamas // EN: *while writing* //, in the sentences:

BG: **Пишейки**, гледах през прозореца.

PL: **Pisząc** patrzyłem w okno.

LT: **Rašydamas** žiūrėjau per langą.

(EN: *While writing, I was looking out of the window.*)

3.2 Participle and verb

It is important to emphasize that participles preserve some properties of the main form of the verb, such as voice, tense and aspect. In Bulgarian, Polish and Lithuanian there are active and passive participles:

a) Present active participle:

BG: говорещ // PL: mówiący // LT: kalbąs / kalbantis // EN: *speaking* // (preserved active voice).

b) Past passive participle:

BG: написан // PL: napisany // LT: parašytas // EN: *written* // (preserved passive voice with information about past tense and perfect aspect of the verbal form).

An interesting fact is that participles preserve the valency properties of the respective verbal form, for instance in Polish and Lithuanian:

PL: Ten mężczyzna zajmuje się drobnym handlem. – Zajmujący się drobnym handlem mężczyzna.

LT: Tas vyras užsiima mažmenine prekyba. – Mažmenine prekyba užsiimantis vyras.

(EN: *This man deals in retail. – A man dealing in retail.*)

The phrase ‘deals in what? / dealing in what?’ requires the instrumental case in Polish and Lithuanian². The valence of the Polish and Lithuanian participle is the same as the valence of the finite verb form.

A comparison of the three languages shows that in Bulgarian a subordinate clause in past perfect tense corresponds to a participle construction in Polish and Lithuanian:

BG: След като си беше написал домашното, той започна да чете книга.

PL: Odrobiwszy lekcje zaczął czytać książkę.

LT: Paruošęs pamokas pradėjo skaityti knygą.

(EN: *Having written his homework, he started reading a book.*)

Polish has a more modest stock of verbal forms with temporal meaning than Bulgarian or Lithuanian. In any case when the lexical means modifying the temporal meanings are taken into account, the participles, and verbal nouns, it is clear that Polish can express also the same temporal meanings.

3.3 Features of the adjective

Adjectives in Polish and Lithuanian can be declined for gender, number and case (in Bulgarian only for gender and number), but do not express a temporal or aspect relation on their own, unlike the participle. These arguments show that participles deserve a separate treatment from adjectives. The main

² This does not apply to Bulgarian which lacks a case paradigm for nouns.

grammatical meaning of the adjective is the attributive meaning. Unlike the participle, which is closely related to a verbal action (state or event in the past, present and future), the adjective denotes a constant property or quality of the object such as:

малко дете | małe dziecko | mažas vaikas // *a little child* //

The adjectives across all three languages function not only as attribute, but also as predicate. As predicate they are only a nominal part of the predicate and express neither time nor aspect. Examples:

Малка къща | Mały dom | Mažas namas // *a small house* //

Къщата е малка. | Dom jest mały | Namas mažas³. (rarely: Namas yra mažas.) // *The house is small.* //

The neuter forms of Lithuanian adjectives possess a semi-predicative function:

LT: Man skanu (adjective, neuter).

BG: На мен ми е вкусно (adverb)./ Вкусно (adverb) ми е.

PL: Smakuje (verb) mi (to).

(EN: *I find it delicious.*)

LT: Gera (adjective, neuter) gyventi kaime.

BG: На село се живее добре (adverb).

PL: Dobrze (adverb) się mieszka na wsi.

(EN: *Living in the village is good.*)

Our observations show that participles have to be considered apart from the adjectives, since adjectives do not carry the verbal characteristics: voice, tense, aspect and valence. Mixing adjectives and participles is a sign of insufficient knowledge of the grammatical structure of Slavic or Baltic languages. Unification of adjectives and participles might be allowed for languages without aspect and/or whose descriptive system of aspect and tense of the verbal form is simpler compared to that of Slavic or Baltic languages. That is the main reason why participles have to be classified as separate POS and not re-qualified as adjectives.

4. Towards development of annotated trilingual electronic resources

Morphosyntactic descriptions for Bulgarian have been developed in several projects, the first of which are for the purposes of corpora processing at the morpho-lexical level in MTE project of EC. The MTE consortium developed morphosyntactic specifications and word-form lexical lists (so called lexicons) covering at least the words appearing in the MTE corpus. For each of the six MTE languages, a lexical list containing at least 15,000 lemmata was developed for use with the morphological analyzer. Each lexicon entry includes information about the inflected-form, lemma, POS, and morphosyntactic specifications. A mapping from the morphosyntactic information contained in the lexicon to a set of corpus tags (used by the POS disambiguator) was also provided, according to the MULTTEXT tagging model. The structure of the lexicon entry is the following:

word-form <TAB> **lemma** <TAB> **MSD** <TAB> **comments**

where **word-form** represents an inflected form of the lemma, characterised by a combination of feature values encoded by **MSD**-code (**MSD**: **M**orpho**S**yntactic **D**escription); the fourth (optional) column, comments, is currently ignored and may contain either comments or information processable by other tools.

Here is an excerpt from the Bulgarian lexicon:

³ In Lithuanian the word order plays a great role in distinguishing the two functions.

потвърждение = Ncns-n

потвърждението потвърждение Ncns-y

потвърждения потвърждение Ncnp-n

потвържденията потвърждение Ncnp-y

(потвърждение: *confirmation, corroboration*).

The **MSDs** are provided as strings, using a linear encoding; an efficient and compact way for the representation of the flat attribute-value matrices. In this notation, the position in a string of characters corresponds to an attribute, and specific characters in each position indicate the value for the corresponding attribute. That is, the positions in a string of characters are numbered 0, 1, 2, etc., and are used in the following way: the character at position 0 encodes part-of-speech; each character at position 1, 2, ..., n , encodes the value of one attribute (person, gender, number, etc.), using the one-character code; if an attribute does not apply, the corresponding position in the string contains the special marker “-” (hyphen). By convention, trailing hyphens are not included in the **MSDs**. Such specifications provide a simple and compact encoding, and are similar to feature-structure encoding used in unification-based grammar formalisms. When the word form is the very lemma, then the equal sign is written in the lemma field of the entry (“=”).

For Bulgarian the morphosyntactic descriptions were designed on the basis of the traditional POS classification according to the traditional Bulgarian grammar (Bulgarian Grammar 1993). Each word form is assigned a label encoding the major category (POS), type where applicable (e.g., proper *versus* common noun) and inflectional features. Punctuation is also included, as are abbreviations, numbers written in digits, and unidentified objects (residuals).

The morphosyntactic descriptions for Polish: the description of Polish by Saloni [16] serves as a basis for the morphosyntactic descriptions for Polish and has been adapted to a large degree to the MTE MSD format in [15].

The system of morphosyntactic tags developed for the Polish at the Institute of Computer Science, Polish Academy of Sciences (IPI PAN), is based on a sound methodological foundation comprising linguistic work by authors such as J.S.Bień, Z.Saloni, M.Świdziński. It is thanks to this foundation that the IPI PAN’s tagset goes beyond the fossilised traditional framework dating back to Aristotle. On the other hand, the MTE tagset, which serves as a point of reference here, is based on the traditional subdivision into parts of speech (this is why, among others, pronouns have been singled out as a part of speech).

Consequently, the aim of our work is neither to revise the good and highly refined IPI PAN tagset nor to replace it with a new tagset for Polish. The issue in question is what kind of compromise should be sought when developing a joint tagset to be used for simultaneous description of the three languages in the BG-PL-LT parallel corpus. For some reasons the MTE tagset (developed previously for many languages) has been selected as the leading one for this corpus. Therefore, the aim of our work is to provide a theoretical study of various categories of Polish (and Lithuanian), to set priorities (e.g. morphological, semantic, syntactic) in identifying various meanings and to provide a classification of morphosyntactic phenomena which does not contradict the MTE standard and does not deviate too strongly from the IPI PAN tagset.

It cannot be excluded that due to the obvious difficulties in achieving consistency of the intertagset the BG-PL-LT corpus will use the IPI PAN tagset for Polish and its modification for Lithuanian. This solution would certainly necessitate a list of more or less close equivalents for the two tagsets: a tagset for Bulgarian on the one hand, and the IPI PAN tagset on the other (for Polish and an extended version for Lithuanian).

It is important to emphasise that only a coherent tagset for a parallel multilingual corpus 1) allows complete linguistic confrontation, 2) enables identification of linguistic facts, 3) enables a search based on pre-defined unambiguous morphosyntactic characteristics.

The morphosyntactic descriptions for Lithuanian: as a basis for morphosyntactic descriptions of Lithuanian serve the Academic grammar of the Lithuanian language [12] and the Functional grammar of Lithuanian [17]. A tool for morphosyntactic annotation for Lithuanian - *MorfoLema* - has been created by Vytautas Zinkevičius in Centre of Computational Linguistics of Vytautas Magnus University (Lithuania) [19]. The program *MorfoLema* can perform a morphosyntactic analysis and generate forms of Lithuanian words based on user's morphosyntactic characteristic. For disambiguation the *MorfoLema* uses „Two-level morphology" method of Kimmo Koskenniemi [11].

The next step of the development of a system for morphological annotation (*Morfologinis anotatorius* [21]) has been realised by Vidas Daudaravičius and Erika Rimkutė. Vidas Daudaravičius has created disambiguation tools for the *Morfologinis anotatorius*. More information about the *Morfologinis anotatorius* and used set of tags we can find on [21] in Lithuanian (the names of tags are in Lithuanian, because the authors of the *Morfologinis anotatorius* didn't use English terms). It is possible to perform online a morphosyntactic analysis through the web-page [22]. The results are visualized on the screen, and it is possible to receive the result as a file.

The authors of the Lithuanian *Morfologinis anotatorius* (see [21]) use the traditional to Lithuanian description of POS. They add two new POS: acronym (like LR for *Lietuvos Respublika* 'Republic of Lithuania') and abbreviation (like gen. for *generalinis* 'main, leading (chief)'). In practice these are not POS, but a means to denote some phenomenon specific to the written language.

The list of POS used for Lithuanian in *Morfologinis anotatorius* follows:

	POS	LT term	LT acronym
1.	noun	daiktavardis	dkt.
2.	adjective	būdvardis	bdv.
3.	numeral	skaitvardis	sktv.
4.	pronoun	įvardis	įv.
5.	verb	veiksmažodis	vksm.
6.	adverb	prieveiksmis	prv.
7.	interjections	jaustukas	jst.
8.	onomatopoeic words	ištiktukas	išt.
9.	particles	dalelytė	dll.
10.	prepositions	prielinksnis	prl.
11.	conjunctions	jungtukas	jng.
12.	acronym	akronimas	akronim.
13.	abbreviation	sutrumpinimas	sutr.

Subcategories such as gender, number, case, present, past, passive, active, etc., are described as separate categories and are not related to POS. This division is in correspondence with many of the subcategories in the Lithuanian academic grammar.

There are certain differences, for example: new case illative (who into? what into? where to?), new gender: bendroji giminė (bi-gendered), new number dviskaita (dual number), new voice reikiamybės (lat. necessitatis, eng. necessity). The grammar recognizes only synthetic verb tenses and adds one form of past tense būtaisis laikas (lat. praeteritum, eng. past). The authors of *Morfologinis anotatorius* deviate from the tradition and ascribe the *tense* characteristic to participles, do not distinguish the analytic tense forms (for example, present perfect, present inchoative), but describe every element of theirs separately. They also form new categories: stabiliosios frazės (phrasal expressions), romėniški skaičiai (roman number), teigiamumas, negiamumas (negation, confirmation), apibrėžtumas (definiteness/indefiniteness). The category of apibrėžtumas (definiteness/indefiniteness) has two subcategories: įvardžiutinis (definiteness) and neįvardžiutinis (indefiniteness).

The names of tags are in Lithuanian, because the authors of the *Morfologinis anotatorius* did not use English terms.

The tag list for Polish and Lithuanian, based on [13], [14], [18], [21], [22] and used in the example below, follows:

For Polish:

acc – accusative	m3 – masculine 3
adj – adjective	n – neuter
conj – conjugation	nom – nominative
dat – dative	pl – plurale
f – feminine	perf – perfective
gen – genitive	pos – positive degree
inf – infinitive	praet – past
interp – punctuation mark	prep – preposition
m1 – masculine 1	sg – singular
m2 – masculine 2	subst – noun

For Lithuanian:

3 asm. – 3rd person	prv. – adverb
būt. k. l. – past	sep. – punctuation mark
dkt. – nomen	teig. – confirmation
dlv. – participle	tiesiog. n. – indicative mood
N. – dative	veik. r – active voice
neįvardž. – indefiniteness	vyr. g. – masculine
nelygin. l. – positive degree	vksm. – verb
nesngr. – non-reflexive	vns. – singular
nežinomas – unknown	V. – nominative

A comparison between experimental annotations of the following sentence “*For answer Gandalf cried aloud to his horse.*”⁴ of the parallel corpus was performed:

BG: Вместо отговор Гандалф гръмогласно подвикна на коня си:
 PL: Zamiast odpowiedzieć hobbitowi, Gandalf krzyknął głośno do swego wierzchowca:
 LT: Užuot atsakęs Gendalfas garsiai riktelėjo žirgui:

The annotation of the Bulgarian text is done with MTE MSDs, and ISSCO TAGGER [20] is used for disambiguation. For manual annotation of the Polish and Lithuanian text the above-mentioned descriptors are used, because these languages lack developed MTE language specifications. Establishing a 1-1-correspondence between the tags used and the MTE tagset does not present an insurmountable difficulty. The result could be seen in **Appendix**.

5. Applications of the trilingual corpus

A parallel corpus of two Slavic languages and one Baltic language is of great interest from the viewpoint of describing the similarities and differences of the formal means of these three languages. Bulgarian belongs to the South subgroup, Polish – to the West subgroup of the Slavic languages. Lithuanian belongs to the Eastern Baltic group. All three languages preserve the special features for each corresponding group. Each one of the three languages however has specific traits which make it unique within the respective language group.

We studied some characteristics in the previous parts. Here we will consider some significant differences between the languages which can be illustrated by examples of texts from the trilingual corpus.

A significant feature is the analytic character of Bulgarian, and the synthetic character of Lithuanian (with some analytic character, like word order in absolute constructions) and Polish. Bulgarian exhibits several linguistic innovations in comparison to the other Slavic languages (a rich system of verbal forms, a definite article), and has a grammatical structure closer to English, Modern Greek, or the Neo-Latin languages than Polish. The definite article in Bulgarian is postpositive, whereas in Lithuanian a similar function is served by qualitative adjectives and adjectival participial forms, both with pronominal declension. Bulgarian preserves some vestiges of case forms in the pronoun system. Polish and Lithuanian exhibit all features of synthetic languages (a very rich case paradigm for nouns). Although Lithuanian has lost the neuter gender of nouns, its case system is richer than the Polish one. Bulgarian and Lithuanian have a high number of verbal forms, but Polish has reduced most of the forms for past tense. Both Polish and Bulgarian have a strongly developed category of verbal aspect. In Lithuanian the verb can have more than one aspect depending on the usage of a base stem for present, past and future tense.

Furthermore, a trilingual corpus can find applications into the design and development of LDB of future bilingual dictionaries, for example, of a LDB supporting a BG–LT dictionary, based on a LDB that supports a BG–PL online dictionary. The advantage of processing a trilingual parallel corpus is to obtain context specific information about syntactic and semantic structures and usage of words in given language or languages. Let us consider an entry of the BG–PL LDB, whose respective dictionary entry of the BG–PL printed dictionary is:

сп|я, -иш *vi.* spać; ~и ми се chce mi się spać, ogarnia mnie senność

The grammatical features of this Bulgarian verb **спя** /sleep/ are:

aspect - imperfect (progressive) /*несвършен вид*/, this verb is **intransitive** /*непребоден*/, its conjugation is a **II type** /*II спрежение*/.

⁴ Tolkien, J.R.R. The Lord of the Rings. Boston : Houghton Mifflin, 1994, p. 731.

Its structure in **BG-PL** LDB is:

```

<entry>
<hw>сп|я'</hw>
<pos>verb</pos>
<gram>imperfect</gram>
  <conjugation><orth>-и'</orth>
    <type>II</type>
  </conjugation>
<subc>intransitive</subc>
<struc type="Sense" n="1">
<trans> spać </trans>
</struc>
  <struc type="Derivation" n="1">
    <orth>~и ми се</orth>
    <struc type="Sense" n="1">
<trans> chce mi się spać </trans>
<alt><trans> ogarnia mnie senność </trans></alt>
</struc>
</struc>
</entry>

```

A possible structure in a future **BG-LT** LDB should be:

```

<entry>
<hw>сп|я'</hw>
<pos>verb</pos>
<gram>imperfect</gram>
<conjugation><orth>-и'и</orth>
  <type>II</type>
</conjugation>
<subc>intransitive</subc>
<struc type="Sense" n="1">
<trans> miegoti </trans>
</struc>
  <struc type="Derivation" n="1">
    <orth>~и ми се</orth>
    <struc type="Sense" n="1">
<trans> (aš) noriu miego </trans>
</struc>
</struc>
</entry>

```

In conclusion we note that the parallel BG–PL–LT corpus will enrich and uncover some unstudied features of the three languages. It will be useful to linguists-researchers for research purposes alike, for instance in contrastive studies of the three languages together or in pairs.

Besides, the trilingual corpus can be used in education, in schools as well as universities in foreign-language instruction.

References

- [1] Bulgarian Grammar. (1993). Главна редакция Д. Тилков, Ст. Стоянов, К. Попов. Граматика на съвременния български книжовен език. Том 2 / МОРФОЛОГИЯ. Издателство на БАН. София. (In Bulgarian).
- [2] Dimitrova, L., Erjavec, T., Ide, N., Kaalep, H.-J., Petkevic, V., and Tufis, D. (1998). Multext-East: Parallel and Comparable Corpora and Lexicons for Six Central and Eastern European Languages. In: *Proceedings of COLING-ACL '98*. Montréal, Québec, Canada, pp. 315-319.
- [3] Dimitrova, L., Panova, R., Dutsova, R. (2009). Lexical Database of the Experimental Bulgarian-Polish online Dictionary. In: *Metalanguage and Encoding scheme Design for Digital Lexicography. Proceedings of the MONDILEX Third Open Workshop*, Bratislava, Slovak Republic, 15–16 April 2009. 36-47. ISBN 978-5-9900813-6-9.
- [4] Dimitrova, L., V. Koseska-Toszewa. (2008). Some Problems in Multilingual Digital Dictionaries. In: *International Journal Études Cognitives*. 8, SOW, 237–254.
- [5] Dimitrova, L., V. Koseska-Toszewa. (2009). Bulgarian-Polish Corpus. In: *International Journal Cognitive Studies / Études Cognitives*. 9, SOW, (in print).
- [6] May Fan, Xu Xunfeng. (2002). An evaluation of an online bilingual corpus for the self-learning of legal English. http://langbank.engl.polyu.edu.hk/corpus/bili_legal.html
- [7] Ide, N., and Véronis, J. (1994). Multext (multilingual tools and corpora). In *COLING '94*, pages 90-96, Kyoto, Japan.
- [8] Ide, N. (1998). Corpus Encoding Standard: SGML Guidelines for Encoding Linguistic Corpora. *Proceedings of the First International Language Resources and Evaluation Conference*, Granada, Spain, 463-70.
- [9] Gamper, Dongilli. (1999). Primary Data Encoding of a Bilingual Corpus. <http://titus.uni-frankfurt.de/curric/gldv99/paper/gamper/gamperx.pdf>
- [10] Geoffrey Leech. (2004). Developing Linguistic Corpora: a Guide to Good Practice Adding Linguistic Annotation. <http://ahds.ac.uk/guides/linguistic-corpora/chapter2.htm>
- [11] Kimmo Koskeniemi. (1983) Two-level morphology: a general computational model for word-form recognition and production. Publication No. 11. Helsinki: University of Helsinki, Department of General Linguistics.
- [12] Lithuanian Grammar. (1997). Ed. Vytautas Ambrazas, Baltos lankos, Vilnius, pp.802.
- [13] Piasecki, M. (2007). Polish Tagger TaKIPI: Rule Based Construction and Optimisation. *Task Quarterly*. 11, p. 151-167
- [14] Przepiórkowski A. (2004), The IPI PAN Corpus: Preliminary version. Institute of Computer Science, Polish Academy of Sciences, Warsaw.
- [15] Roszko, R. (2009). Morphosyntactic Specifications for Polish. Theoretical foundations. In: *Metalanguage and Encoding Scheme Design for Digital Lexicography. Proceedings of the MONDILEX Third Open Workshop*, 15-16 April 2009, Bratislava. 140–150. ISBN 978-80-7399-745-8.
- [16] Saloni, Z., W. Gruszczyński, M. Woliński, R. Wołosz (2007). Słownik gramatyczny języka polskiego, Wiedza Powszechna, Warszawa, CD + 177 s. (In Polish)
- [17] Valeckienė, A. (1998). Funkcinė lietuvių kalbos gramatika, Mokslo ir enciklopedijų leidybos institutas, Vilnius, pp.415. (In Lithuanian)
- [18] Woliński, M. (2003). *System znaczników morfosyntaktycznych w korpusie IPI PAN*, Polonica, XXII-XXIII, p. 39-55 (In Polish)
- [19] Zinkevičius, V. (2000). Lemuoklis - morfologinei analizei. *Darbai ir dienos*, 24, Vytauto Didžiojo universitetas, p. 245-274 (In Lithuanian).
- [20] ISSCO TAGGER: <http://www.issco.unige.ch/staff/robert/tatoo/tagger.html#design>
- [21] Morfologinis anotatorius (tagger for Lithuanian): http://donelaitis.vdu.lt/main.php?id=4&nr=7_1
- [22] http://donelaitis.vdu.lt/main.php?id=4&nr=7_2
- [23] ParaSol corpus: http://www.uni-regensburg.de/Fakultaeten/phil_Fak_IV/Slavistik/RPC/

Appendix

Bulgarian (MTE annotation):

BG: Вместо отговор Гандалф гръмогласно подвижна на коня си:

```

<cesAna version="1.0" type="lex disamb">
<chunkList>
<chunk type="s">
  <tok type=WORD>
    <orth>Вместо </orth>
    <disamb><base>вместо</base><ctag>RG</ctag></disamb>
    <lex><base>вместо</base><msd>Rg</msd><ctag>RG</ctag></lex>
    <lex><base>вместо</base><msd>Sp</msd><ctag>SP</ctag></lex>
  </tok>
  <tok type=WORD>
    <orth>отговор</orth>
    <disamb><base>отговор</base><ctag>NCMS-N</ctag></disamb>
    <lex><base>отговор</base><msd>Ncms-n</msd><ctag>NCMS-N</ctag></lex>
  </tok>
  <tok type=WORD>
    <orth>Гандалф</orth>
    <disamb><base>Гандалф</base><ctag>NPMS-N</ctag></disamb>
    <lex><base>Гандалф</base><msd>Npms-n</msd><ctag>NPMS-N</ctag></lex>
  </tok>
  <tok type=WORD>
    <orth>гръмогласно</orth>
    <disamb><base>гръмогласно</base><ctag>RA</ctag></disamb>
    <lex><base>гръмогласен</base><msd>A--ns-n</msd><ctag>ANS</ctag></lex>
    <lex><base>гръмогласно</base><msd>Ra</msd><ctag>RA</ctag></lex>
  </tok>
  <tok type=WORD>
    <orth>подвижна</orth>
    <disamb><base>подвижна</base><ctag>VMIA3S</ctag></disamb>
    <lex><base>подвижна</base><msd>Vmia2s</msd><ctag>VMIA2S</ctag></lex>
    <lex><base>подвижна</base><msd>Vmia3s</msd><ctag>VMIA3S</ctag></lex>
    <lex><base>подвижна</base><msd>Vmip1s</msd><ctag>VMIP1S</ctag></lex>
  </tok>
  <tok type=WORD>
    <orth>на</orth>
    <disamb><base>на</base><ctag>SP</ctag></disamb>
    <lex><base>на</base><msd>Qgs</msd><ctag>QG</ctag></lex>
    <lex><base>на</base><msd>Sp</msd><ctag>SP</ctag></lex>
  </tok>
  <tok type=WORD>
    <orth>коня</orth>
    <disamb><base>кон</base><ctag>NCMS-S</ctag></disamb>
    <lex><base>кон</base><msd>Ncms-s</msd><ctag>NCMS-S</ctag></lex>
  </tok>
  <tok type=WORD>
    <orth>коня</orth>
    <disamb><base>кон</base><ctag>NCMT</ctag></disamb>
    <lex><base>кон</base><msd>Ncmt</msd><ctag>NCMT</ctag></lex>
  </tok>
  <tok type=WORD>
    <orth>си</orth>
    <disamb><base>си</base><ctag>PX</ctag></disamb>
    <lex><base>си</base><msd>Ncns-n</msd><ctag>NCNS-N</ctag></lex>
    <lex><base>си</base><msd>Px-----ys</msd><ctag>PX</ctag></lex>
    <lex><base>си</base><msd>Px---d--yp</msd><ctag>PX</ctag></lex>
    <lex><base>си</base><msd>Qgs</msd><ctag>QG</ctag></lex>
    <lex><base>си</base><msd>Qvs</msd><ctag>QV</ctag></lex>
    <lex><base>съм</base><msd>Vaip2s</msd><ctag>VAIP2S</ctag></lex>
  </tok>

```

```

<tok type=PUNCT><orth>:</orth><ctag>PERIOD</ctag></tok>
</chunk>
</chunkList>
</cesAna>

```

Polish [13]

PL: Zamiast odpowiedzieć hobbitowi, Gandalf krzyknął głośno do swego wierzchowca:

```

<!DOCTYPE cesAna SYSTEM "xcesAnaPI.dtd">
<cesAna version="1.0" type="lex disamb">
<chunkList>
<chunk type="s">
<tok>
    <orth>Zamiast</orth>
    <lex disamb="1"><base>zamiast</base><ctag>prep:gen</ctag></lex>
    <lex><base>zamiast</base><ctag>conj</ctag></lex>
</tok>
<tok>
    <orth>odpowiedzieć</orth>
    <lex disamb="1"><base>odpowiedzieć</base><ctag>inf.perf</ctag></lex>
</tok>
<tok>
    <orth>hobbitowi</orth>
    <lex disamb="1"><base>hobbit</base><ctag>subst.sg.dat.m3</ctag></lex>
    <lex><base>hobbitowy</base><ctag>adj.pl.nom.m1:pos</ctag></lex>
</tok>
<ns/>
<tok>
    <orth>,</orth>
    <lex disamb="1"><base>,</base><ctag>interp</ctag></lex>
</tok>
<tok>
    <orth>Gandalf</orth>
    <lex disamb="1"><base>gandalfa</base><ctag>subst.pl.gen.f</ctag></lex>
</tok>
<tok>
    <orth>krzyknął</orth>
    <lex disamb="1"><base>krzyknąć</base><ctag>praet.sg.m1:perf</ctag></lex>
    <lex><base>krzyknąć</base><ctag>praet.sg.m2:perf</ctag></lex>
    <lex><base>krzyknąć</base><ctag>praet.sg.m3:perf</ctag></lex>
</tok>
<tok>
    <orth>głośno</orth>
    <lex disamb="1"><base>głośno</base><ctag>adv.pos</ctag></lex>
</tok>
<tok>
    <orth>do</orth>
    <lex disamb="1"><base>do</base><ctag>prep:gen</ctag></lex>
</tok>
<tok>
    <orth>swego</orth>
    <lex><base>swój</base><ctag>adj.sg.gen.m1:pos</ctag></lex>
    <lex disamb="1"><base>swój</base><ctag>adj.sg.gen.m2:pos</ctag></lex>
    <lex><base>swój</base><ctag>adj.sg.gen.m3:pos</ctag></lex>
    <lex><base>swój</base><ctag>adj.sg.gen.n:pos</ctag></lex>
    <lex><base>swój</base><ctag>adj.sg.acc.m1:pos</ctag></lex>
    <lex><base>swój</base><ctag>adj.sg.acc.m2:pos</ctag></lex>
</tok>
<tok>
    <orth>wierzchowca</orth>
    <lex disamb="1"><base>wierzchowiec</base><ctag>subst.sg.gen.m2</ctag></lex>
    <lex><base>wierzchowiec</base><ctag>subst.sg.acc.m2</ctag></lex>
</tok>
</chunk>
</chunkList>
</cesAna>

```

Lithuanian

LT: Užuoat atsakęs Gendalfas garsiai riktelėjo žirgui:

LT version [22]:

```
<word="Užuoat" lemma="užuoat" type="prv., teig., nelygin. I.">
<space>
<word="atsakęs" lemma="atsakyti(-o,-ė)" type="dlv., teig., nesngr., veik. r, būt. k. I., neįvardž., vyr. g., vns., V.">
<space>
<word="Gendalfas" lemma="Gendalfas" type="nežinomas">
<space>
<word="garsiai" lemma="garsiai" type="prv., teig., nelygin. I.">
<space>
<word="riktelėjo" lemma="riktelti(-telia,-telėjo)" type="vksm., teig., nesngr., tiesiog. n., būt. k. I., vns., 3 asm.">
<space>
<word="žirgui" lemma="žirgas" type="dkt., vyr. g., vns., N.">
<sep=":">
```

Future and Possibility*

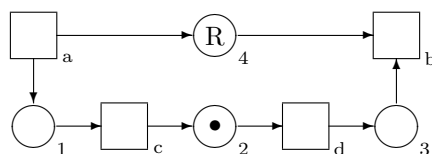
Violetta Koseska¹ and Antoni Mazurkiewicz

¹ Institute of Slavistics of PAS, Warsaw

² Institute of Computer Science of PAS, Warsaw

A single net can represent more than one history; a history is a possible course of actions and states described by the supporting net. Transitions (actions) and places (states) can occur within a history several times, if they are contained in a cycle of the net; therefore, we should speak about element occurrences rather than about elements themselves. A history supported by a net has the following properties. Firstly, it contains the state of utterance. Secondly, if it contains an event, it contains also all its preconditions and all its postconditions, as indicated by the supporting net. Thirdly, if a history contains a state, then it contains at most one event initiating it and at most one event terminating it, if such events do exist. A history is complete, if it cannot be extended by adding new objects. A complete history of a net containing a cycle can be infinite; in such cases we frequently use a partial history. Below we listed net schemes of chosen real situations related to the future and to different aspects of possibilities.

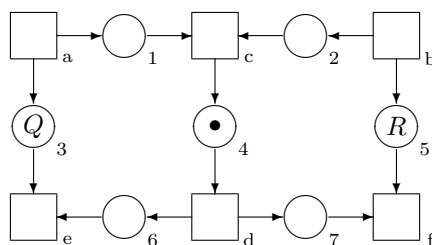
Scheme 1: Present



Ivan sega chete vestnik (BG)
 Now, John is reading a newspaper (GB)
 Jan teraz czyta gazetę (PL)
 Ivan sejchas chitaet gazetę (RU)

(2)	Teraz	Now
(4)	Czytanie	Reading

Scheme 2: Present, with 3 independent states

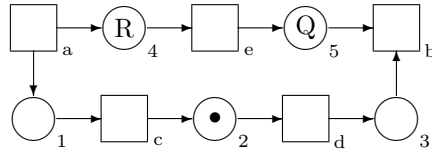


Ivan e bolen i lezhi v bolnicata (BG)
 John is ill and he is in hospital (GB)
 Jan leży chory i jest w szpitalu (PL)
 Ivan bolen i lezhit w bolnice (RU)

* Work supported by EU FP7 project GA211938 MONDILEX "Conceptual Modelling of Networking of Centres for High-Quality Research in Slavic Lexicography and Their Digital Resources".

(3) jest chory	is ill
(5) jest w szpitalu	is in hospital
(4) teraz	now
(a) początek choroby	begin of illness
(b) początek pobytu w szpitalu	begin of being in hospital
(a) koniec choroby	end of illness
(b) koniec pobytu w szpitalu	end of being in hospital

Scheme 3: Present, with 2 consecutive states - limited knowledge possibility
(deterministic, no braching at states)



Ako toj sega e vyv vlaka, utre shte byde vyv Varshava;
ako sega e vyv Varshava, vchera e trjabvalo da byde vyv vlaka (BG)

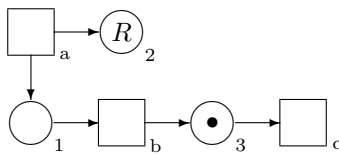
If he is in a train now, he will be in Warsaw tomorrow;
if now he is in Warsaw, he had to be in a train yesterday (GB)

Jeśli jest teraz w pociągu, jutro będzie w Warszawie;
jeśli jest teraz w Warszawie, wczoraj musiał być w pociągu (PL)

Esli sejchas on w poezde, to zavtra budet w Varshave;
esli sejchas on w Varshave, to vchera on dolzhen byl byt' w poezde (RU)

(2) Teraz	Now
(4) W pociągu	In a train
(5) W Warszawie	In Warsaw

Scheme 4: Possible present, known past



Vyzmoznno e da e vse oshte v otpuska (BG)

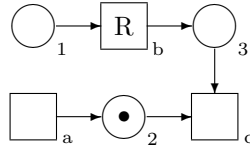
Maybe he is still on holidays (GB)

Możliwe, że on jeszcze jest na wakacjach (PL)

Vozmoznno, chto on eshche v otpuske (RU)

(2) Jest na wakacjach	he is on holidays now
(3) Teraz	Now
(a) Początek wakacji	beginning of holidays

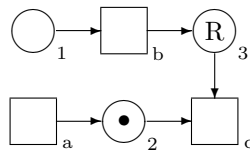
Scheme 5: possible event - indeterministic possibility



Vyzmozhno e, che negovata otpuska e svyrshila (BG)
 Possibly he has finished his holidays by now (GB)
 Możliwe, że skończył już swój urlop (PL)
 Vozmozhno, chto otpusk ego uzhe zakonchilsja (RU)

(1)	Jest na wakacjach	he is on holidays
(2)	Teraz	Now
(b)	Koniec urlopu	End of holidays

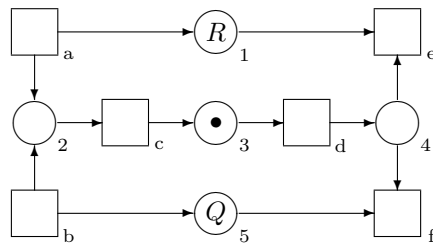
Scheme 6: Present, possible past



Vyzmozhno e, toj da e veche sled otpuskata si (BG)
 Possibly he is out of holidays now (GB)
 Możliwe, że jest on już po urlopie (PL)
 Vozmozhno, chto on uzhe posle otpuska (RU)

(1)	Jest na wakacjach	he is on holidays
(2)	Teraz	Now
(3)	Po urlopie	Out of holidays

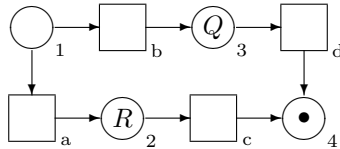
Scheme 7: Indeterministic mutual exclusion
 (different termination of states - branching at states)



Sega Marija e ili v Krakov, ili v Gdansk (BG)
 Now, Mary is either in Cracow or in Gdansk (GB)
 Teraz Maria jest albo w Krakowie, albo w Gdańsku (PL)
 Seichas Marija ili w Krakove, ili v Gdanske (RU)

(1) Maria jest w Krakowie	Mary is in Cracow
(3) Teraz	Now
(5) Maria jest w Gdańsku	Mary is in Gdansk
(a) Początek pobytu w Krakowie	Beginning of stay in Cracow
(b) Początek pobytu w Gdańsku	Beginning of stay in Gdansk
(e) Koniec pobytu w Krakowie	End of stay in Cracow
(f) Koniec pobytu w Gdańsku	End of stay in Gdansk

Scheme 8: Conditional past



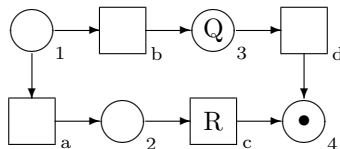
Ako toj beshe kazal istinata, tja ne e bila tam (BG)
 If he told the truth, she wasn't there (GB)
 Jeśli on mówił prawdę, to jej tam nie było (PL)
 Esli on govoril pravdu, to ee tam ne bylo (RU)

Another version:

Ako tja e bila tam, toj lyzhe (BG)
 If she was there, he is lying (GB)
 Jeśli ona tam była, on kłamie (PL)
 Esli ona tam byla, to on lzhet (RU)

(2) On mówi prawdę	he is telling the truth
(3) Ona tam jest	She is over there
(c) Powiedział prawdę	He told the truth

Scheme 9: Conditional event



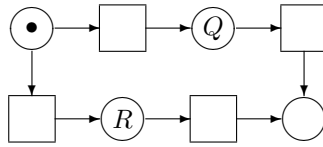
Ako toj e kazal istinata, tja ne e bila tam (BG)
 If he told the truth, she wasn't there (GB)
 Jeśli on powiedział prawdę, to jej tam nie było (PL)
 Esli on skazal pravdu, to ee tam ne bylo (RU)

Alternatively:

Ako tja e bila tam, toj e izlygal (BG)
 If she is there, he lyied (GB)
 Jeśli ona tam była, on skłamał (PL)
 Esli ona tam byla, on solgal (RU)

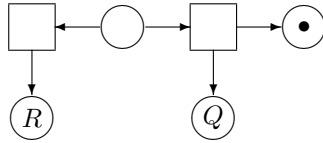
- | | | |
|-----|-------------------|-------------------------|
| (2) | On mówi prawdę | he is telling the truth |
| (3) | Ona tam jest | She is over there |
| (c) | Powiedział prawdę | He told the truth |

Scheme 10: future possibility



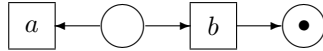
Surely, it will be either R or Q

Scheme 11: future possibility



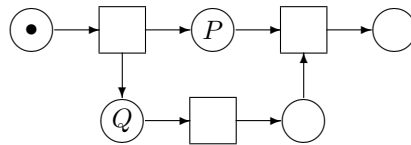
It could be R , but Q is; Q is, but it could be R
 Now you are well (Q), but you could be dead (R)

Scheme 12: Past possibility



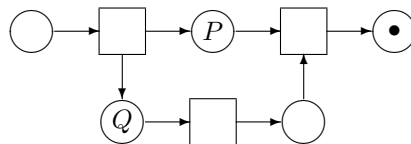
It could happen a , but b has happened
 I could hit you (a), but I didn't do that (b)

Scheme 13: Positive future

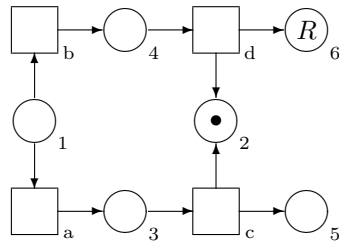


Q will be terminated by Friday (P)

Scheme 14: Positive present

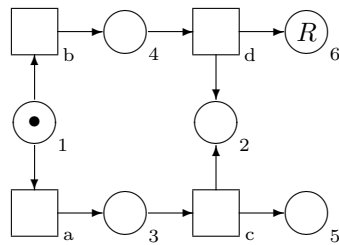


Q has been terminated on Friday(P)

Scheme 15: Conditional possibility in the past

Ako beshe vchera pozvynila, shtjah dnes da dojda (BG)
 If you called me yesterday, I would come today(GB)
 Gdybyś wczoraj zadzwoniła, to bym dzisiaj przyszedł (PL)
 Esli by ty vchera pozvonila, to ja by segodnja prishel (RU)

(1) Wczorajsza decyzja	Yesterday's decision
(2) Teraz	Now
(3) Dzwoni	She is calling
(4) Nie dzwoni	She is not calling
(5) Przychodzę	I am coming
(6) Nie przychodzę	I am not coming

Scheme 16: Conditional possibility in the future

Ako mi se obadish, utre shte dojda (BG)
 If you call me, I will come tomorrow (GB)
 Jeśli zadzwonisz, to jutro przyjdę (PL)
 Esli ty pozvonish, zavtra ja pridu (RU)

(1) Dzisiejsza decyzja	Today's decision
(2) Jutro	Tomorrow
(3) Dzwoni	She is calling
(4) Nie dzwoni	She is not calling
(5) Przychodzę	I am coming
(6) Nie przychodzę	I am not coming

List of situations with uncertainty caused by a local view

Scheme	Situation	Temporal meaning
Scheme 1		Present
Scheme 2		Present, with two independent states
Scheme 3		Present, with two consecutive states
Scheme 4		Certainly past, possibly present
Scheme 5		Event that possibly has already happened
Scheme 6		Possibly past, certainly present

List of situations with structural possibilities

Scheme	Situation	Temporal meaning
Scheme 7		Two present mutually exclusive states
Scheme 10		Possibility in the future
Scheme 13		Perfective future
Scheme 14		Perfective past

List of possibilities induced by a choice

Scheme	Situation	Temporal meaning
Scheme 11		Conditional possibility in the past
Scheme 12		Not possible conditionality
Scheme 15		Condition not realized in the past
Scheme 16		Condition for the future
Scheme 8		Possibility of past states
Scheme 9		Possibility of past events

Conclusions.

In natural languages the semantic category of time, presented in this paper, is formed by a combination of states and events together with states of utterance supplemented, if necessary, with some additional indications making clear the intention of the speaker. While the past and present tenses serve to inform the reader/leastener about events that took place already and the speaker is sure about their truth or falsehood, the future tense does not supply us with such knowledge. In effect the nets described in the present paper, mainly describing future tense and its relationship to past and present, together with the phenomena of indeterminism, are different than those described in previous papers.

Due to the net formalism one can easily and simply describe all that happens during the state of utterance and after it. The net description makes clear that not only the precedence - succession relation is fundamental for expression the time in natural languages, but also the simultaneity and concurrency relations are equally important and play an important part in understanding natural language sentences.

Our knowledge is changing and increasing; this process is accompanied with changing the relation of the speaker to the described reality. Before Copernicus the statement "*the Earth goes round the sun*" was false, but nowadays it is true. The form of future tense expresses the laws of the nature, but is it really a "future tense"?

The meaning of the form of *futurum* in sentences like (1) *Sun will set at 5 pm* has not the same meaning as in the sentence (2) *John will be at home tomorrow*. The meaning of (1) one can call the universal - general, since the sun in this season always sets at 5 pm, and apparently is related to the general meaning of *praesens*, see (Koseska, Gargov (1990)). Natural language sentences that describe laws of nature can be viewed as unquestionably true. However, from the natural language point of view, the *futurum* form has not the meaning of future tense. The cause of future states and events occurring after the utterance state exists prior to the utterance state, at the utterance state, and after it. Therefore, the meaning of such sentences does not depend on the utterance state and can be considered as timeless and universal. Using forms of *futurum* has nothing to do with the future tense.

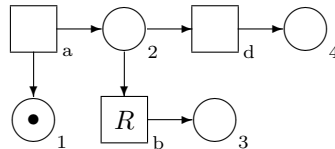
In natural languages, difference between past and future arises from the relation earlier - later on one hand, and from the position of the state of utterance with respect to described situation on the other hand. The net description shows clearly the states and events following the utterance state and, at the same time, which of them will occur in effect of a choice between two or more possible events. Such a choice is necessary for the proper understanding the speaker intensions and it has **to be consistent with the amount of knowledge of the speaking subject** (see schemes 5 and 6).

It is worthwhile to note the description of conditional sentences. In the present paper, for the sake of clarity and precision we use syntactic means in the form of compound statements with subordinate clauses and construction 'if ... then ...' (schemes 2,8,9). The special attention should be paid to the net description given in scheme 15. Bulgarian form of positive perfective *futurum praeteriti* (*futurum exactum praeteriti*) of perfective verbs '*shtjah da dojda*' occurs in a compound sentence, where there is lack of other forms expressing conditionality. With similar situation we have in English, where the '*would*' form serves to express conditionality. In Polish and Russian the similar role play expressions '*gdybyś/jeśli byś*'. Bulgarian form '*shtjah da dojda*' explicitly expresses that the referred event did not happened. The meaning of Bulgarian *futurum praeteriti* is stressing the lack of a state (an event) after the utterance state. The nets, respecting non-linearity of succession, turn out to be a handy tool for describing such situations.

In scheme 9 a net without actual possibility is presented; more precisely, it describes a past possibility which had been resolved in one way: *'If you stay closer, you will not be alive now'*. It shows that discussing hypothetic and unreality in the framework of conditionality has been not justified on semantic level.

Consider scheme 17.

Scheme 17: Possibility at present



Mozhesh da go ubiesh, be, chovek! (BG)

You can kill him, man! (GB)

Możesz go zabić, człowieku! (PL)

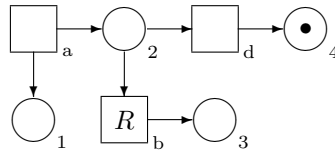
Mozhesh jego ubit'! (RU)

(4) he is alive

(3) he is dead

and scheme 18

Scheme 18: Possibility at present, irrational (?)



Shteshe da go ubiesh, be chovek! (BG)

You nearly killed him, man! (GB)

O mało go nie zabiłeś, człowieku! (PL)

Ty mog jego ubit'! (RU)

(4) He is alive

(3) He is dead

Each potential sentence is hypothetical, but not the other way round. This observation leads to a restriction of the conditionality notion in natural languages. It is especially significant for the conditionality theory, justified in details in the Bulgarian / Polish Grammar volume, dedicated to the conditional modality in Bulgarian and Polish.

References

1. Koseska-Toszewa V., Gargov, G.: *Bylgarsko-polska sypostavitelna gramatika*, vol. 2, Semantichnata kategorija opredelenost/neopredelenost, Sofija, 1990
2. Koseska-Toszewa V., Maldzieva, V., Pencev, J.: *Gramatyka konfrontatywna bulgarsko-polska*, t.6, cz 1, Modalność. Teoretyczne problemy opisu. Warszawa 1996
3. Koseska-Toszewa, *Semantyczna kategoria czasu*, GKBP, SOW, Warszawa, 2007
4. Koseska V., Mazurkiewicz A.: *Net representation of sentences in natural languages*, Advances in Petri Nets, 1988, LNCS 340, Springer Verlag, pp 249-259
5. Koseska V., Mazurkiewicz A.: *Net Net Based Description of Modality in Natural Language (on the Example of Conditional Modality)*, Proc. of the MONDILEX Second Open Workshop, Kiev,(2008) (to appear)
6. Mazurkiewicz, A.: *A Formal Description of Temporality (Petri Net approach)*, *Lexicographic tools and techniques*, Proc. of the MONDILEX First Open Workshop, Moscow, ISBN 978-5-990813 (2008) pp 98-108
7. Petri, C.A.: *Fundamentals of the Theory of Asynchronous Information Flow*, Proc. of IFIP'62 Congress, 1962, North Holland Publ. Comp., pp 386-390
8. Reichenbach, H.: *Elements of Symbolic Logic*, New York, McMillan Publ. (1944)

Theory of Lexicographic Systems. Part 3

Volodymyr Shyrokov

Ukrainian Lingua-Information Fund, National Academy of Sciences of Ukraine

Abstract. The theory of lexicographic systems is applied to establish the structure of the L-system for the explanatory Ukrainian Language Dictionary (ULD). The conceptual model for ULD-20 (Ukrainian Language Dictionary in 20 volumes) is worked out. The hidden symmetry of ULD-20 is stated. The technology of parsing of the ULD-11 and automatically forming the lexicographic database of the explanatory Ukrainian Language Dictionary is described. The computer instrumental system as a kind of the virtual lexicographic laboratory for working out the explanatory Ukrainian Language Dictionary is presented.

Keywords: Ukrainian Language Dictionary, dictionary parsing, lexicographic structure, lexicographic database, hidden symmetry of the lexicographic system, semantic state.

9. The Ukrainian Language Dictionary in 11 Volumes (1970-1980)

The construction of the lexicographic system for the Ukrainian Language Dictionary (ULD) and creation of the lexicographic database (LDB) and computer technology for making the explanatory dictionaries on the basis of this dictionary are considered in this section. The application of the theory of lexicographic systems allowed carrying out the parsing of the dictionary (conversion of the dictionary text into LDB) in the automatic mode for a very complex lexicographic object that is ULD. It also allowed constructing the effective computer technology for making the explanatory dictionaries.

The fundamental academic edition “Ukrainian Language Dictionary” in 11 volumes [1] is deservedly considered to be the highest achievement of the Ukrainian national lexicography. It is a normative and reference lexicographic work of the explanatory type that meets the basic requirements of the lexicography and covers the Ukrainian vocabulary from the time of I. Kotlyarevsky to the 1970s. Based on the considerable and diverse lexical-phraseological material in terms of origin and functioning, the ULD has the register of over 134 thousand words. The significance of the 11-volume Ukrainian Language Dictionary for the linguistic science is determined by three main factors.

Firstly, it is a result of the Ukrainian linguistics development in which the achievements of the linguistic theory and the practice of several generations of the Ukrainian scientists are concentrated.

Secondly, it represents the actual base for the new linguistic researches. In particular, the major directions of its research are the analysis of the dictionary register from the grammatical view, stylistic differentiation as well as characteristics of the main parameters for lexicographic interpretation of the vocabulary, search for the ways to improve the vocabulary fixation and for the methods and the ways of presenting semantic structures and relations.

Thirdly, new editions of the explanatory dictionaries that have appeared in the last decade, are also based methodologically on the ULD. In fact, they are its “clones”. At least we could not find there any fundamental lexicographic innovations despite of sometimes considerable and often unjustified expansion of the register.

However, by the end of the 1990s the ULD-11 has been outdated by its content. Thus, there is a necessity of creating a new updated version of the Ukrainian Language Dictionary approximately in 20 volumes (ULD-20). The new version of the ULD from the outset was focused on the modern computer technology, since such a work should necessary have a digital version. Such a formulation of the problem raised a number of new tasks in linguistics and in the systems engineering.

First of all, the basic linguistic principles and directions of the ULD-11 modernization of the ULD-20 compilation were worked out produced. They are:

1. Maximum complete saving and using the theoretical and practical heritage of the national lexicography embodied in the ULD-11 structure and corpus;
2. Removing the rudiments of the totalitarian regime from the ULD; deideologizing the lexicographic material (removing quotations from the works of Marxist-Leninist orientation, removing the vocabulary that has lost its sense, clearing the semantics from the ideological influence of the previous epoch);

3. Considerable expanding the lexical and text-illustrative base of the Dictionary (maximum complete presenting the vocabulary and phraseology of the Holy Scripture; the significant increase of the list of authors cited by attracting works of the previously forbidden writers and the writers of the newest era; creating the Ukrainian National Linguistic Corpus and using of its resources; using the language resources of the Internet);

4. Returning specific features to the Ukrainian language (coordinating the lexicographic material with grammatical and stylistic rules of the modern Ukrainian literary language, including the extraction of the implicit Russianisms; attracting the lexicographic material, which reflects the national and historical realities and mental concepts of the Ukrainian people);

5. Reflecting the lexical-semantic dynamics which reflects the social and historical context of Ukraine's development (formation of Ukraine as an independent state; changing social order; democratization of society; scientific and technological revolution; transition to the information society and the knowledge society; changing political situation in the country; growing political and economic activity in all segments of the Ukrainian society; globalization; expansion and intensification of the international relations of Ukraine);

6. Updating the lexicographic material;

7. Filling the lexical lacunas.

The linguistic instructions mentioned above and a very short term for creating the ULD-20 have stipulated for a necessity of creating special computer environment that would be able to be an effective instrument when compiling the Dictionary. The digital text of ULD-11 had become the factual basis of such environment, for which it was necessary to develop a formal structure of its L-system, because the elements of the formal structure were identified with the relevant elements of the LDB structure allowing its formation in the automatic mode.

With this approach, the structural aspect of ULD acquires a decisive importance, because the success of the ULD-20 project depends on its implementation. Despite the fact that linguists have written a lot about the structure of dictionaries in general and about the ULD structure, but before our works on the theory of lexicographic systems these structures were considered mainly as means that help making convenient and compact presentation of many linguistic facts in one object – dictionary entry. In contrast to this pragmatic approach, the theory of lexicographic systems provides the conceptual apparatus and formal means for considering the lexicographic structures as free-standing linguistic and information objects. They allow the abstraction from the specific text realizations in the dictionary entries, and being presented in the form of some formal objects they acquire an independent linguistic interpretation and can be used for entirely new linguistic researches. Such researches using formal representatives of the ULD lexicographic structures were held in ULIF. The new linguistic results were obtained. In our opinion, it is almost impossible to achieve them by other methods. Just using the apparatus of the theory of lexicographic systems allowed separating the lexicographic structures from the dictionary text and looking at them as at the original demonstrators of language system, in which certain regularities inherent to language were “encrypted”. To justify the stated above let us go on to the establishment and analysis of the ULD lexicographic structures.

10. Presenting Semantic Structures of ULD by Means of the Theory of Lexicographic Systems

The modelling of the Ukrainian Language Dictionary by means of the theory of lexicographic systems is probably the most serious test for this theory because the ULD is a dictionary of the integral type that contains a large number of linguistic phenomena, and its volume allows considering that these phenomena are presented in their representative version. The construction of formal lexicographic structures of the ULD, accomplished in the early 1990s, significantly strengthened our belief in the accuracy of the concept of lexicographic systems and led to deepening of its content in the line of greater rigor and transparency of the basic principles of the theory.

Let us present the ULD lexicographic structure according to the methodology of the theory of L-systems.

The lexicographic effect of the special type takes place in the middle of the speech stream. According to the general instructions of the theory of lexicographic systems, the result of this effect is that a generation of a set of elementary information units (EIU) appears. It is qualified as *a class of words of the Ukrainian language* – $I^w(U)$. This qualification, however, can not serve as a definition of this set, because more

accurate descriptions of the lexical units is needed for the goals of lexicography. To determine these descriptions it is necessary to attract linguistic concepts related to the structure of words, to attribution of the meanings of certain grammatical categories to them, to the origin (etymology), to the concepts of sense and meaning, to the functioning in the contexts etc. For this purpose, the following types of descriptions for this object are determined in the conceptual model of the lexicographic system:

$$\begin{array}{ll}
 V_f[I^w(U)] - \text{phonetic}; & V_{\text{Sem}}[I^w(U)] - \text{semantic}; \\
 V_G[I^w(U)] - \text{graphic}; & V_{\text{Et}}[I^w(U)] - \text{etymological}; \\
 V_M[I^w(U)] - \text{morphemic}; & V_{\text{Styl}}[I^w(U)] - \text{stylistic}; \\
 V_{\text{gr}}[I^w(U)] - \text{grammatical}; & V_{\text{st}}[I^w(U)] - \text{statistical etc.}
 \end{array} \quad (3.1)$$

Each of these descriptions or any combination of them may form an independent L-system. The construction of mappings for these L-systems to certain standard data models is also possible. For example, the morphemic description $V_{\text{Morph}}[I^w(U)]$ in a certain approximation may be presented in terms of the relational data model. The set of attributes (names of attributes) in this case may be defined as following:

$$\Pi := \text{„prefix“}; K := \text{„root“}; C := \text{„suffix“}; \quad (3.2)$$

M := „interfix“; Φ := „inflection“

Let us mark the appropriate domains, domain elements and the names of information variables as:

$$\begin{array}{ll}
 D(\Pi) & = \{\Pi_1, \Pi_2, \dots, \Pi_k\} \\
 D(K) & = \{K_1, K_2, \dots, K_p\} \\
 D(C) & = \{C_1, C_2, \dots, C_s\} \\
 D(M) & = \{M_1, M_2, \dots, M_m\} \\
 D(\Phi) & = \{\Phi_1, \Phi_2, \dots, \Phi_f\}
 \end{array} \quad (3.3)$$

A certain aggregate of functions (relations of the conceptual model) is determined on the Cartesian product of domains $D(\Pi) \times D(K) \times D(C) \times D(M) \times D(\Phi)$:

$$\{r(\Pi, K, C, M, \Phi)\} \quad (3.4)$$

A set of values for this model is identified with a certain own subset of the set $I^w(U)$. They will be, in fact, real words in the language with the morphemic analysis, the algorithms of which are realized with a set of relations (3.4). Thus, the relations $r(\Pi, K, C, M, \Phi)$ (actually, they are rules of word formation) set the law of composing elements from $D(\Pi) \times D(K) \times D(C) \times D(M) \times D(\Phi)$, so that a sequence:

$$\Pi_9 * K_9 * C_1 * M_1 * \Phi_1, \quad (3.5)$$

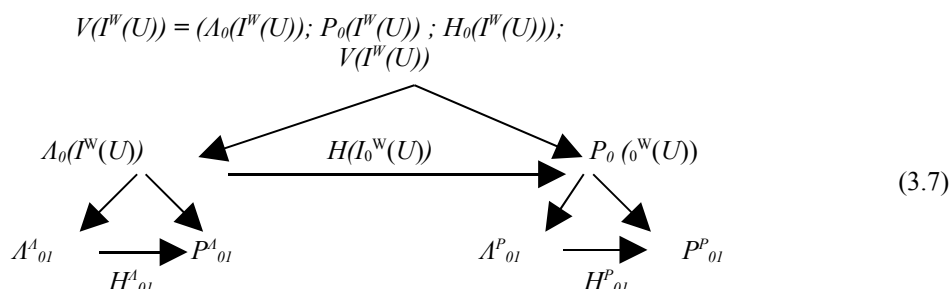
where concatenation is marked with an asterisk, represents a real word of language. The structure of the relevant elementary L-system is determined here with such a diagram:

$$\begin{array}{c}
 V_{\text{Morph}}[I^w(U)] \\
 \swarrow \quad \searrow \\
 \begin{array}{cc}
 \leftarrow H \equiv \{r(\Pi, K, C, M, \Phi)\} \rightarrow & (3.6) \\
 \leftarrow A[I^w(U)] \equiv D(\Pi) \times D(K) \times D(C) \times D(M) \times D(\Phi) & \rightarrow P[I^w(U)] \equiv \{\Pi_9 * K_9 * C_1 * M_1 * \Phi_1\}
 \end{array}
 \end{array}$$

The statement in this case is simplified and serves only as illustration.

Let us assume that a determinative characteristic of systems forming for the linguistic complex “Word” is a value property, which in turn is a bearer of the relation of form and content. The formal part of this relation, correlative with a value of complex EIU “Word”, is concentrated around its properties in the *system of language*. While the substantial part of the same relation is concentrated mainly around the ontological dimensions of this complex which concern to being as is and are the result of the fundamental property of words to represent the reality objective reality.

This results in the fact that L-system with the class EIU “Word” should support the structure of the recursive reduction of minimum second order, so it looks like:



Let us mark a word class of the Ukrainian language with $I^W(U)$ character on the diagram (3.7); the natural interpretation of the structural element $A_0(I^W(U))$ is its interpretation as a carrier of the grammatical semantics, and $P_0(I^W(U))$ — of the lexical semantics, respectively. The function $H_0(I^W(U))$ provides their connection and combination of the linguistic object in a unit.

In turn, the structural element $A_0(I^W(U))$, considered as a representative of the grammatical semantics, acquires interpretation as L-system that supports the relations of inflection and word forming.

A set of semantic relations is presented in the construction of lexical semantics representative $P_0(I^W(U))$. A hierarchy of lexical meanings (for each lexeme x they are concentrated in the structural element $A^P_{0i}(I^W(U))$ and presented as interpretation formulas or dictionary definitions) and relevant microcontexts (examples of word usage), presented in the elements $P^P_{0i}(I^W(U))$, is selected. A certain part of identifiers for the relations of synonymy, antonymy, meronymy, hyponymy, hyperonymy etc. is concentrated in the elements $A^P_{0i}(I^W(U))$.

According to the general construction of L-system, a set of $V(I^W(U))$ descriptions for the class of elementary information units are selected in the structure of the explanatory L-system. The analogue of these units is a set of dictionary entries, a set of relations between certain subsets of the set of dictionary entries, and between separate subsets of the structural elements of dictionary entries.

The structure of the explanatory L-system may be detailed in different ways:

1) to find out, which types of the lexicographic effects should be presented in the dictionary; to analyze the ways and develop the procedure of their presenting as elementary L-systems; to apply the procedure of integration to them;

2) to apply successively the procedure of recursive reduction to the elementary L-system presented with upper level of the diagram (3.7);

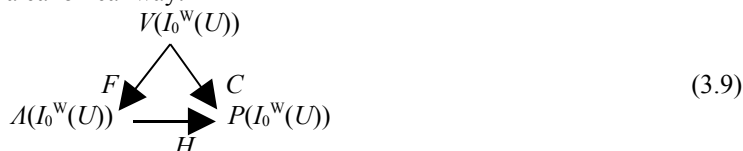
3) to construct the structure of the explanatory L-system ab origin as an elementary L-system.

The most consistent and correct way is the first one, but it is the most difficult. The second way is close to it. Here the construction of the elementary explanatory L-system is the simplest way, so let us use it. It doesn't eliminate the application of the procedure of recursive reduction later, but to certain elements of the structure constructed using the third way of the explanatory L-system.

According to the general methodology, let us mark the ULD L-system with $V(I_0^W(U))$ character, where EIU class $I_0^W(U)$ is an aggregate of the Ukrainian words in the initial form, and $V(I_0^W(U))$ is a set of their descriptions:

$$V(I_0^W(U)) = \cup_{x \in I_0^W(U)} V(x), \quad (3.8)$$

where $V(x)$ is interpreted as a dictionary entry with a register word x . The centrality of the dictionary entry in the explanatory dictionary structure is reflected just in this way. Then the main relation – relation of form and content – is presented in a canonical way:



$F V(I_0^W(U)) = A(I_0^W(U)); C V(I_0^W(U)) = P(I_0^W(U)); A(I_0^W(U)) \cap P(I_0^W(U)) = \emptyset,$
 $H \circ F = C$, where the following combination of mappings is marked with “ \circ ” symbol;

$$A(I_0^W(U)) = \cup_{x \in I_0^W(U)} A(x) \quad ; \quad P(I_0^W(U)) = \cup_{x \in I_0^W(U)} P(x) \quad (3.10)$$

The macrostructures:

$$F \sigma[\beta] = \lambda[\beta] \quad \text{and} \quad C \sigma[\beta] = \rho[\beta] \quad (3.11)$$

and the relevant microstructures:

$$\lambda[\beta] |_{V(x)} \equiv \lambda(x); \quad \rho[\beta] |_{V(x)} \equiv \rho(x) \quad (3.12)$$

are induced on $A(I_0^W(U))$ and $P(I_0^W(U))$ as limitations $\lambda[\beta]$ and $\rho[\beta]$ on $V(x)$. The maximum lexical-grammatical and lexical-semantic information is concentrated in the structural elements $\lambda(x)$ and $\rho(x)$, so the construction of their explicit view is the main problem of the explanatory lexicography.

Let us see the processes and results of constructing the ULD microstructures within certain lexical-grammatical classes.

11. The Microstructure of the Register Parts of the ULD Dictionary Entries

11.1. Verb. Structure

Constructing the formal representatives of the left part structure of the ULD dictionary entries $A(x)$ is carried out separately for each part of speech, because they are different for different parts of speech. Let us construct the formal representatives of $A(x)$, when x runs over a set of verbs of the ULD. When constructing the formal structures $A(x)$ of the verb, we have found a new linguistic phenomenon, which can be described as a “hidden symmetry of the ULD L-system.”

The concept of symmetry in general plays an important role in science. A large number of works is devoted to this phenomenon. The works devoted to phenomenology and metaphysics of symmetry, particularly in areas not related to natural science are published nowadays. At the same time a scientific paradigm based on the knowledge and study of the role of different types of symmetry in nature, has formed in the natural sciences (particularly in the XX century). H. Lorentz, A. Einstein, H. Poincaré investigated the fundamental role of symmetry in constructing the physical space and time, and in the structure of possible interaction of the physical objects, what favoured the creation of relativistic worldview. The identification of the so-called internal symmetries of elementary particles has led to discovery of a number of subatomic and subnuclear particles of matter, and to construction of their classification. So a set of useful concepts both in practical and in epistemological aspects was formulated (dynamic symmetry, broken symmetry etc.). Formulating the gauge theory has led to the concept of gauge symmetry and the construction of a number of theories that combine various interactions, which at first seemed to be totally unrelated, in a single mathematical scheme (in the first place, the electromagnetic and weak, and then – in the so-called Standard Model – electromagnetic, weak and strong; the theories of Grand Unification based on the concept of supersymmetry, seeking to unite all the known types of interactions into a comprehensive whole, are developed intensively).

A characteristic feature of the specified theories, using the concept of symmetry, is a high level of their formalization, which involves a very sophisticated mathematical apparatus (different Lie algebras and groups, the superalgebras and supergroups, their various representations, etc.).

Unfortunately, the stated can not be yet fully extended to the linguistic science, though even here the concept of symmetry (as well as the concept of asymmetry) is used for a long time. This is a result of the difficulties, which the processes of formalization and mathematization in linguistics are faced with. But more interesting are the cases where we are able to establish the formal regularities in the language system and to draw the quite definite conclusions from them regarding the properties of its symmetry.

This section is devoted to exposition of the example of this kind. Its conceptual basis is a phenomenon of so-called hidden symmetry typical for a number of other (non-language) systems. But still there are no standard, simple and reliable methods to identify the type of system symmetry and its formalization in science. Therefore, in many cases, the system symmetry is not evident at all, it seems to be “hidden” from the observer, and its identification to a large degree is a result of intuition and perhaps good luck of the researcher, particularly in cases when he works with the poorly structured and poorly formalized material.

Such material is predominantly the language material.

The phenomenon, which we qualify as a demonstration of hidden symmetry of the language system was discovered while studying the ULD-11 structure. Unlike many other linguistic objects, a dictionary is a well-structured and rather formalized material. Therefore, at first glance, it might seem that finding in it any symmetry is not so complicated and interesting task. However, as you will see below, the study of the ULD-11 structure has led to the discovery of such its hidden symmetry, which is already quite strict law for the entire system of the Ukrainian language, not only for the linguistic material presented in the ULD-11. Moreover, the type of the law established is new, because nothing of the kind was not still found among the investigated and established regularities in the Ukrainian language.

Let us analyze the structure of verb dictionary entries of the ULD.

For example, the left part « $L(X) = \text{ЗВИВАТИ}$ і рідко ІЗВИВАТИ , аю, аеш, недок., ЗВІТИ , зів'ю, зів'еш і рідко ІЗВІТИ , ізів'ю, ізів'еш і діал. ЗВІНУТИ і рідко ІЗВІНУТИ , ну, неш, док., *перех.*» of the dictionary entry with the register word « ЗВИВАТИ » has a register row consisting of six components: ЗВИВАТИ ; ІЗВИВАТИ ; ЗВІТИ ; ІЗВІТИ ; ЗВІНУТИ ; ІЗВІНУТИ . They are united with lexical semantics common to all the components of the row expressed with a complex of lexical meanings common to them:

« $P(X) = 1$. Скручуючи, спітаючи нитки, стебла і т. ін., робити, виготовляти що-небудь. *Торік бувало тут, над сим [цим] потоком, Звивала я тобі вінки барвисті (Леся Українка); Гніздо звивають ластівки під дахом (М. Рильський); Дід звивав шнурок (І. Франко); Кожній [дівці] хочеться, щоб їй дружки .. весільних пісень просівали.., щоб їй вильце звили (Грицько Григоренко); Гей на Івана, гей на Купала Красна дівчина доли птала; Із барвіночку вінець ізвила, На чисту воду плисти пустила (А. Вахнянин).*

2. Згоргати, скручувати що-небудь у сувій, згорток, кільце. – *Що хочете тим сказати, тату? – спитала вона спокійно, звиваючи назад розсіпані ноти (О. Кобилянська); – Буде буря! – .. На великому чорному баркасі .. справді звивали вітрила (М. Коцюбинський); Порицький помалу розв'язав книжку, звинув у каблучку шнурочок (Леся Українка); // Загоргати, закручувати в що-небудь. – Я тобі звину в папір і в шматину [листа], візьмеш в ремінь, то не згубиш! (І. Франко).*

3. розм. Закручуючи по спіралі, навколо осі, піднімати вгору. *Звиваючи куряву смерчем, промчав вихор (З. Тулуб).*

4. діал. Завивати (волосся). *Кинулась [панночка] на ліжко: – Роззувай! .. А вмієш ти волосся звивати? – питає .. – Я не вмію кіс ізвивати!.. (Марко Вовчок).*

5. діал. Припиняти діяльність. *Рішив звинути всі патрулі, дати Довбушеві можливість проявити себе, аби хоч таким чином напасти на слід (Г. Хоткевич).*»

In this case the first two components of the register row (ЗВИВАТИ ; ІЗВИВАТИ) refer to the imperfective aspect, while the remaining four (ЗВІТИ ; ІЗВІТИ ; ЗВІНУТИ ; ІЗВІНУТИ) – to the perfective aspect.

Such feature of presenting the entries in the ULD in conjunction with the morphological structure of the Ukrainian language leads to a certain symmetry in their structure expressed in the regularities, which will be stated below. While establishing these regularities, we will follow a certain methodology of research, formulated in the form of requirements to the theory, which are as follows:

1. A theory must be based on the minimum possible number of axioms (postulates). Assumptions ad hoc are considered invalid.

2. A theory must satisfactorily explain all the phenomena belonging to the class.

3. A theory must have some predictive power, and correctly predict new phenomena or facts.

We realize that even in these strict requirements to the theory there is certain ineffable inaccuracy that consists of the immanent indefiniteness inherent in the process of constructing dictionary entries and defining a set of lexical meanings. In fact, the authors-compilers of the dictionary entries are the lexicographers that are not free from subjectivism, and in whole a team of lexicographers represents a so-called “collective subject”. Consequently, the semantic complexes, representing lexical semantics of the register rows, should be considered a subjectivity “modulo” of the lexicographic team.

However, even within this subjectivity the structure of verb entries of the ULD obeys some hidden and even unwitting by its authors regularities.

Namely: if we consider the texts $L(X)$, where X runs over a set of verbs of the Ukrainian language, abstracting from their content – as a linear chain of characters, we can see that some “subchain” are selected in the structure of these chains. These subchains look like enclosures, and these enclosures are selected in some invariant way, so that the various subchains do not intersect.

This regularity of the $A(X)$ chains structure is so regular that it allows formulating it even in the axiomatic form, which is formulated in the form of three postulates following the methodological orientations given above.

Postulate 1. Each Ukrainian verb is realized in the language by a lexeme (with a definite and fixed lexical semantics) with one or two values of the category of aspect.

Let us call a chain of characters in $A(X)$, relating to a certain value of the category of aspect¹, an *aspect complex* (or simply a *complex*) and mark it with C_i character, where index i equals 1, if there is only one aspect complex in the dictionary entry $A(X)$, and $i = 1, 2$, if there are two aspect complexes in the dictionary entry $A(X)$.

For example, there is only one aspect complex $C_1 = \text{БУХИКНУТИ, ну, неш}$, in the dictionary entry $A(X) = \text{БУХИКНУТИ, ну, неш, док.}$:

Thus, we have an enclosure of the first level:

$$C_1 \subset A(X) = A(\text{БУХИКНУТИ}).$$

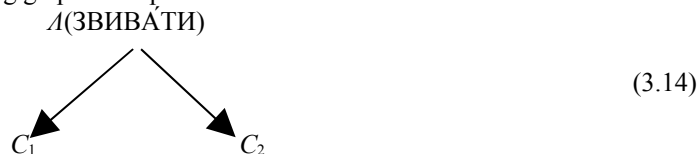
Let us represent this fact in the graphic form:



where the relation of enclosure “ \subset ” is represented with an arrow. Similarly, there are two aspect complexes in the dictionary entry $A(\text{ЗВИВА́ТИ})$ given above:

$$\begin{array}{l} C_1 = \text{ЗВИВА́ТИ } i \text{ рідко } \text{ІЗВИВА́ТИ, аю, аеш}, \\ C_2 = \text{ЗВІ́ТИ, зів'ю́, зів'еш } i \text{ рідко } \text{ІЗВІ́ТИ, ізів'ю́, ізів'еш } i \text{ діал. ЗВІ́НУТИ } i \text{ рідко } \text{ІЗВІ́НУТИ, ну,} \\ \text{неш.} \end{array}$$

Thus the complex C_1 refers to the imperfective aspect, and complex C_2 – to the perfective aspect. This linguistic fact, obviously, has the following graphical representation:



Postulate 2. Each verb lexeme with a definite and fixed lexical semantics and a specific aspect value can be realized with verbs belonging to no more than three different inflection classes.

The inflection class identifiers in the ULD are the paradigmatic indicators, which typically are sets of quasiinflections of the first and second person, singular, present tense. If they are not enough, then some other forms of inflection for the lexeme are given – anyway, a set of paradigmatic indicators is chosen sufficient to identify the inflectional paradigm for a particular lexeme.

Let us call a chain (a dictionary entry fragment within a single aspect complex), relating to one inflectional paradigm, a *paradigmatic block* or simply *block*. We will mark the blocks with B_{ik} character, where the first character $i = 1, 2$, is a complex number, inside of which the block is located, and the second character $k = 1, 2, 3$, enumerates the blocks within a complex.

Let us consider a block structure of the dictionary entry $A(\text{ЗВИВА́ТИ})$ as an example. In the first complex: $C_1 = \text{ЗВИВА́ТИ } i \text{ рідко } \text{ІЗВИВА́ТИ, аю, аеш}$, there is only one block in it:

$$B_{11} = \text{ЗВИВА́ТИ } i \text{ рідко } \text{ІЗВИВА́ТИ,}$$

which is united by a set of paradigmatic indicators «аю, аеш». In the second complex: $C_2 = \text{ЗВІ́ТИ, зів'ю́, зів'еш } i \text{ рідко } \text{ІЗВІ́ТИ, ізів'ю́, ізів'еш } i \text{ діал. ЗВІ́НУТИ } i \text{ рідко } \text{ІЗВІ́НУТИ, ну, неш}$, there is a maximum possible number of blocks – three, namely:

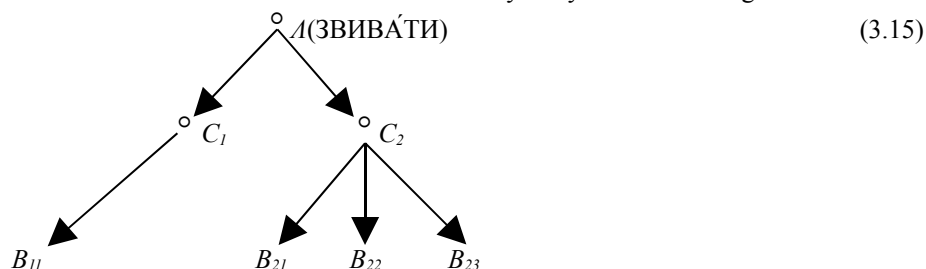
$$B_{21} = \text{ЗВІ́ТИ,}$$

defined by the indicators «зів'ю́, зів'еш»;

¹ There are two aspects for the verbs of the Ukrainian language: perfective («доконаний» – «док.») and imperfective («несовершенный» – «недок.»). There are also so-called bi-aspectual verbs that are marked with marks: *док. і недок., недок. і док.* etc. In accordance with our methodology, these double marks represent individual values of the category of aspect.

defined by the indicators «ізів'ю́, ізів'єш»; and
 $B_{23} = i \text{ діал. ЗВІ́НУТИ } i \text{ рідко ІЗВІ́НУТИ,}$
 united into the block by the indicators «ну, неш».

The graphical representation of the block structure for this dictionary entry is the following:



Postulate 3. Each verb lexeme with a specific aspect value, a certain belonging to a particular inflection class and a definite and fixed lexical semantics can be realized with the verbs that have no more than four phonetic and morphemic variants (as a rule, it is a prefixal and root variation, but also may be a suffixal one).

We will call these phonetic and morphemic variants *components*, and mark them with K_{ijr} , where index $i = 1, 2$, enumerates the aspect complexes, to which the relevant component belongs; $j = 1, 2, 3$, enumerates the blocks within a complex, and $r = 1, 2, 3, 4$, enumerates the components within the relevant block.

To illustrate the effect of these components let us consider the dictionary entry:

$\Lambda(X) = \text{ЗСИХА́ТИ } i \text{ рідко ІЗСИХА́ТИ, аю, а́еш, недок.}, \text{ЗСО́ХНУТИ } i \text{ рідко ІЗСО́ХНУТИ, ЗСО́ХТИ } i \text{ рідко ІЗСО́ХТИ, хну, хнеш; мин. ч. зсох } i \text{ зсо́хнув, ла, ло; док.}$

There are two aspect complexes in it. In the first complex:

$C_1 = \text{ЗСИХА́ТИ } i \text{ рідко ІЗСИХА́ТИ, аю, а́еш,}$

there is only one block:

$B_{11} = \text{ЗСИХА́ТИ } i \text{ рідко ІЗСИХА́ТИ,}$

to which in turn two components belong:

$K_{111} = \text{ЗСИХА́ТИ}; K_{112} = \text{ІЗСИХА́ТИ};$

In the second complex:

$C_2 = \text{ЗСО́ХНУТИ } i \text{ рідко ІЗСО́ХНУТИ, ЗСО́ХТИ } i \text{ рідко ІЗСО́ХТИ, хну, хнеш; мин. ч. зсох } i \text{ зсо́хнув, ла, ло,}$

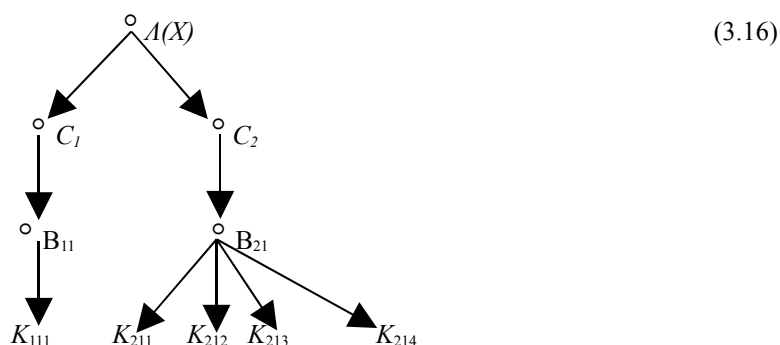
there is only one block too:

$B_{21} = \text{ЗСО́ХНУТИ } i \text{ рідко ІЗСО́ХНУТИ, ЗСО́ХТИ } i \text{ рідко ІЗСО́ХТИ.}$

But in this block there is a maximum number of components – four, namely:

$K_{211} = \text{ЗСО́ХНУТИ}; K_{212} = \text{ІЗСО́ХНУТИ}; K_{213} = \text{ЗСО́ХТИ}; K_{214} = \text{ІЗСО́ХТИ.}$

Graphically, this fact is represented as follows:



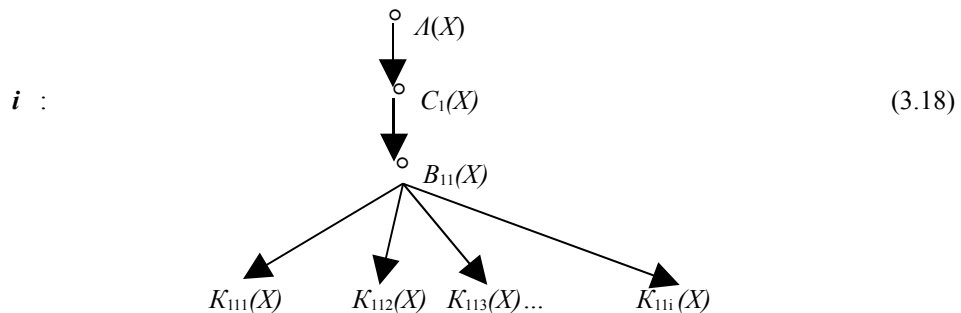
From the described properties of the composition of verbal entries $\Lambda(X)$, the conclusion on the existence of particular linguistic regularity in their structure, expressed formally by the system of enclosures, follows:

$$\Lambda(X) \supset C \supset B \supset K, \quad (3.17)$$

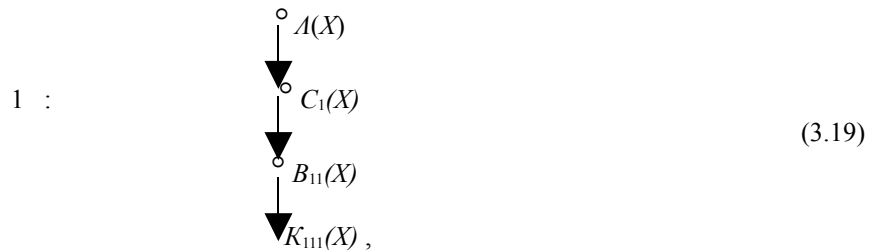
where the set of aspect complexes belonging to this $\Lambda(X)$ is marked with C character (not more than 2

complexes), the set of paradigmatic blocks is marked with B character (not more then 3 blocks in each complex), the set of components is marked with K (not more then 4 components in each block). Let us call this regularity with a rule «1-2-3-4». In this rule various grammatical, structural-morphological and lexical-semantic phenomena, defining the law of filling the structural elements of the left parts for the ULD verb entries, are unified. The rule «1-2-3-4» represents actually a hidden symmetry in the structure of $A(X)$, allowing the lexicographic structures of quite certain types. Therefore it is not accidentally that this fact was unnoticed even for lexicographers – the authors of the ULD conception and structure and its compilers. We found out this phenomenon only after a long, purposeful and conscious study of the ULD lexicographic structures.

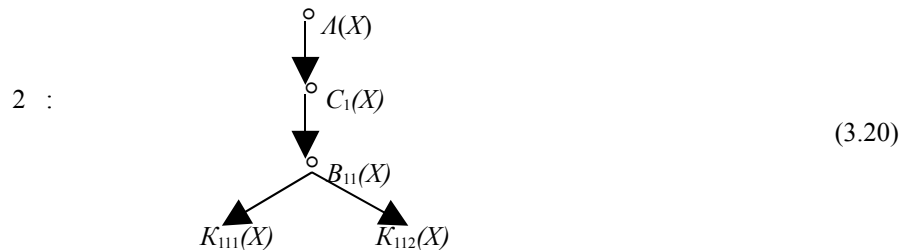
Because the structure of verbal $A(X)$ has a certain symmetry, then there is a way of its formalization, on the basis of which an exhaustive enumeration of permissible lexicographic types of $A(X)$ is possible. To establish and formalize this symmetry, let us introduce some marks. We will mark a set of numbers from 1 to 4 with character I^1 : $I^1 = \{1, 2, 3, 4\}$, and accept that any number i from this set ($i \in I^1$) defines the following graph:



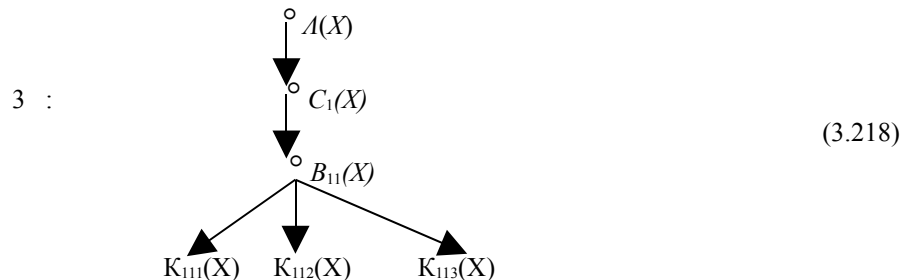
In other words, any number $i \in I^1$ definitely determines a graph allowable by postulates (1-3) and representing the structure $A(X)$, in which there is one aspect complex, one paradigmatic block and i component, $i = 1, 2, 3, 4$. Thus, the number 1 specifies the graph:



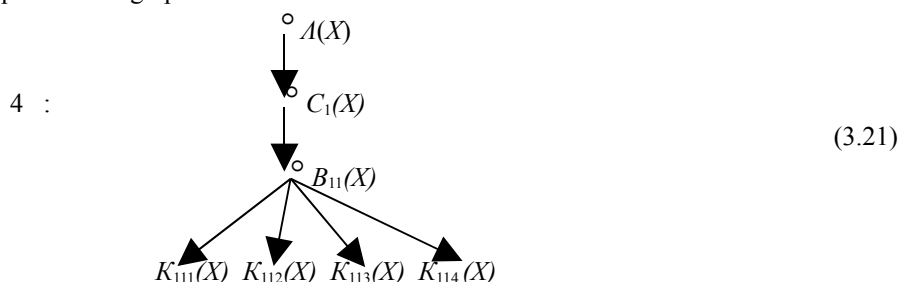
the number 2 specifies the graph:



the number 3 specifies the graph:



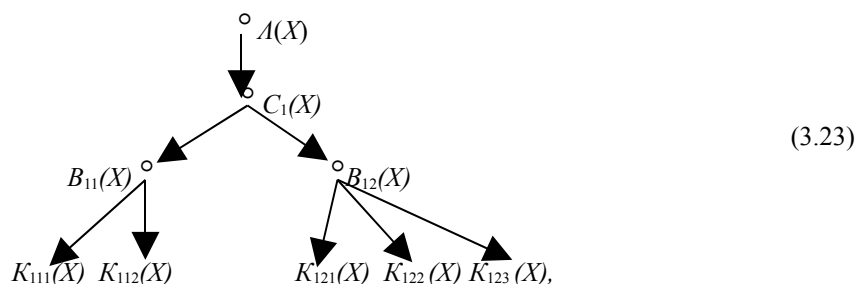
Finally, the number 4 specifies the graph:



These graphs cover all structures of the 1-complex, 1-block verbal entries of the ULD. To describe the rest of permissible structures we will define the Cartesian products:

$$I^1 \times I^1 \times I^2 \text{ and } I^1 \times I^1 \times I^1 \times I^3 \tag{3.22}$$

With the set I^2 consisting of pairs (ij) – the signatures, where the numbers i, j run over the values from 1 to 4 independently, – the structures of 2-blocks complexes are parametrized. Namely, the signature (ij) , $i, j = 1, 2, 3, 4$, specifies the structure of $A(X)$ containing a single complex, which consists of two blocks, the first of which in turn contains i , and the second – j component. For example, the signature $(2\ 3)$ meets the following structural graph:



Similarly, the triple (ijk) , $i, j, k = 1, 2, 3, 4$, specifies the structure of $A(X)$ containing a single complex, which consists of three blocks, the first of which contains i , the second – j , and the third – k component.

1-block (I^1)				2-block (I^2)											
1				11	12	13	14								
2				21	22	23	24								
3				31	32	33	34								
4				41	42	43	44								
3-block (I^3)															
11	11	11	114	211	212	213	214	311	312	313	314	411	412	413	414
1	2	3													
12	12	12	124	221	222	223	224	321	322	323	324	421	422	423	424
1	2	3													
13	13	13	134	231	232	233	234	331	332	333	334	431	432	433	434
1	2	3													
14	14	14	144	241	242	243	244	341	342	343	344	441	442	443	444
1	2	3													

Table 1. Signatures of aspect complexes

Table 1 shows all signatures of the 1-complex lexicographic structures (i) , (ij) , (ijk) . Their total number obviously is: $4 + 16 + 64 = 84$. Each signature corresponds to an unambiguously defined structural graph, which meets the structure of $A(X)$.

We can calculate, by what signatures the verb entries containing a specified number of lexemes in the register row, are realized. It is clear that the maximum possible number of the register row elements for the 1-complex $A(X)$ is 12 (maximum three blocks and maximum four components in each block). The general distribution of signatures on the number of units of the register row is presented in Table 2.

Number of words in the register row of the complex	Signatures, by which the dictionary entries with certain number of the register row components are realized														Number of signatures
	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	
1	1														1
2	2	11													2
3	3	21	12	111											4
4	4	31	22	13	121	112	211								7
5	41	32	23	14	131	122	113	221	212	311					10
6	42	33	24	141	132	123	114	231	222	213	321	312	411		13
7	43	34	142	133	124	241	232	223	214	331	322	313	421	412	14
8	44	143	134	242	233	224	341	332	323	314	431	422	413		13
9	144	243	234	342	333	324	441	432	423	414					10
10	244	343	334	442	433	424									6
11	344	443	434												3
12	444														1
															84

Table 2. The distribution of signatures on the number of units of the register row

With the mechanism described, the structures of $\mathcal{A}(X)$, containing two aspect complexes, are also easily formalized. The structures of these entries are represented by pairs $(\alpha . \beta)$, where α and β independently run over the set of signatures $\{(i), (ij), (ijk)\}$ defined in Table 1.

Maximum possible number of signatures of the type $(\alpha . \beta)$ is 7056 (i.e. 84×84). The total number of structures, satisfying the postulates 1-3, theoretically may be: $7140 = 84 + 7056$.

Thus, the hidden symmetry of the verbal $\mathcal{A}(X)$ leads to the fact that, at first glance, unrestricted number of entry structures is reduced to the final, accurately defined integer.

The formalism constructed, the structure of signatures and their distribution are direct results of the postulates 1–3. These postulates have been formulated as a result of examining the texts of the ULD verbal entries in the 1990s, when the computer database of the Dictionary was not yet formed. Then we asked, whether all verb entries of the ULD satisfy these postulates, and whether there are exceptions to the rule «1-2-3-4.»

It was very difficult to receive the answer to this question at that time because it was simply unreal to view the structures of more than 41 thousand verb entries of the ULD-11 «manually». This problem was solved in 2001, after the ULD database was formed. This allowed creating a tool set for modernizing the Dictionary and holding a number of studies using this tool. Thus, the structures of verbal $\mathcal{A}(X)$ were studied concerning the conformity with the postulates 1-3 and the rule «1-2-3-4». As a result of the computational experiment conducted on the ULD database, only 52 classes of the potentially possible 7140 classes of the verbal $\mathcal{A}(X)$ in the ULD were actually identified. The details of this experiment are described in the monograph [5].

We do not set the goal of holding a thorough linguistic analysis of the system constructed. Let us make just a few comments.

First, the system described determines a certain classification on a set of the Ukrainian verbs. Let us mark the obtained classes of verbs marked with signatures with $\lambda_1, \lambda_2, \dots, \lambda_{52}$, and the sets of verbs belonging to the relevant class – with $q(\lambda_1), q(\lambda_2), \dots, q(\lambda_{52})$. Then it is obvious that:

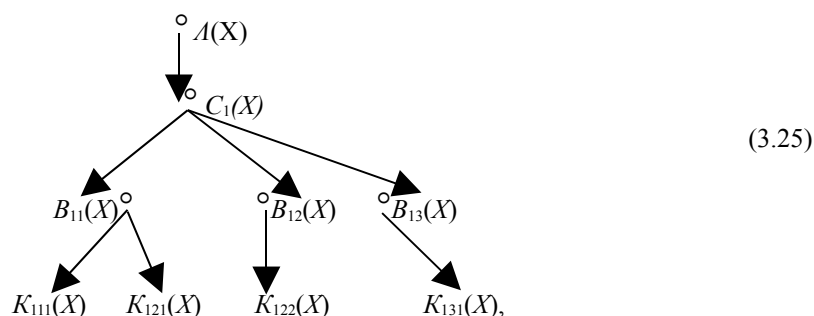
$$q(\lambda_i) \cap q(\lambda_j) = \emptyset \text{ with } i \neq j, \quad (3.24)$$

i.e., any verb can belong to one and only one class. This means that the classification obtained is correct. In addition, it is also quite accurate. No exceptions were found on the array of more than 41 thousand verbal lexemes. Moreover, this classification is also correct for an array of a new 20-volume version of the ULD created in ULIF. There are more than 48 thousand verbs in this array now. Consequently, the classification covers all the facts of the considered class of objects (the Ukrainian verbs) and, therefore, represents a kind of objective law for the whole of the Ukrainian language, not only for a Dictionary.

Let us say a few words about the “empty” structures of the classification, i.e. for which the relevant verbs

in the ULD were not found. We are not worried about this fact. Let us remember, for example, the history of the periodic classification for the chemical elements. The number of cells in it is theoretically not limited, although at the time of D. Mendeleev no more than 63 elements were known and a lot of them were not sufficiently well identified. In the table created by D. Mendeleev there were a number of gaps, but the system – the periodic law – allowed predicting the properties of “missing” elements. Let us note that even in our times only 111 elements are discovered. The latter of them, superheavy, are very unstable and have a very short lifetime; they are not found in the nature and can be received only artificially at the accelerators.

Similarly, as in our case, the existence of certain verbs belonging to the “missing” classes is not eliminated. The availability of free classes gives signals of the word formative potencies of the Ukrainian verb that still have not been used. Perhaps more deep studies and development of the language system will allow identifying such classes, therewith the system “predicts” their morphological properties. For example, among 1-complex signatures identified in the ULD there is a signature (11), which corresponds, for example, to $L(X) = \text{БУЛЬКОТАТИ, очу, очеш } i \text{ БУЛЬКОТИТИ, очу, отиш, } \textit{недож}$. There is a signature (12), which corresponds, for example, to $L(X) = \text{РИБАЛИТИ, лю, лиш, РИБАЛЧИТИ } i \textit{ рідко РИБАЧИТИ, чу, чиш, } \textit{недож}$. But the signature (121) is absent. According to the classification, the signature (121) defines an aspect complex with three paradigmatic blocks with 1, 2 and 3 components each, respectively. Therefore the following graphical representation for the elements of this class is correct:



and, by analogy with the signature (12) variant, the signature (121) corresponds to the following model $L(X)$:

$$\begin{aligned} &R+\text{АЛИТИ, лю, лиш, } R+\text{АЛЧИТИ } i \textit{ рідко } R+\text{АЧИТИ, чу, чиш,} \\ &R+\text{АЧУВАТИ, ую, уєш, } \textit{недож} , \end{aligned} \quad (3.26)$$

where the root of a hypothetical lexeme is marked with R character. Thus, there is a theoretical possibility of developing the lexical system towards the formation of the structural classes still absent in the Ukrainian language, but admissible by its system and its hidden symmetry.

On the other hand, it is intuitively clear that the implementation of the “superheavy” classes, represented by the signatures, for example (344.444) (444.344) (444.444), is hardly probable – the morphological system of the Ukrainian language will not stand them. Consequently, the question on determining the system boundaries arises. The answer to it can be obtained during the complex lexical-grammatical and lexical-semantic studies using the obtained structures of the formal classification for the Ukrainian verb and the lexicographic databases of ULIF.

The next question is whether it is possible to extend the above method of establishing symmetry to other parts of speech of the Ukrainian language. Part of the problem was solved in [3, 7], although there are still several outstanding questions.

Finally, the question arises concerning this problem in other languages, in particular, closely related, for example, in other Slavic languages? It seems that the method described can be applied to these languages, where possibly the existence of some hidden symmetries will also be established.

11.2. The Microstructure of the Interpretation Parts of the ULD Dictionary Entries

In this paragraph we consider the structure of $P(x)$ as a representative of *content* of a language unit in the ULD, where the right parts contain the detailed semantic information about the register units.

The right (interpretation) parts of the ULD entries as a rule are much wider by their sizes than the respective register (left) parts. This is predetermined by the functional and interpretation orientation of this dictionary entry element. In other words, the right part can be considered as a fragment of linguistic text corpus structured in a suitable way. If a set of left parts of the ULD dictionary entries represents the grammatical semantics of the Ukrainian language, then the correspondent set of right parts of the dictionary entries represents a system of lexical semantics. So the amount of the explanatory part is often ten times more than that of the left part. However, the structural organization of the right part is a bit simpler. Without going into the analysis and interpretation of this fact, let us go to the description of the structure elements of $P(x)$.

By analogy with the left part, the right part also consists of several nested substructures, the identification of which is connected with some linguistic phenomena. In the right part $P(x)$ structure of the ULD dictionary entry the effects of the register unit polysemy and its collocations are displayed. The polysemy has a two-stage character and consists of rubrics that represent the lexical meanings of the relevant lexemes and their shades of meaning. All structural effects that we describe here are formally defined in the dictionary text using special punctuation marks, position, bolding, etc. Therefore the construction of the formal representative of structure can be completely algorithmized.

The analysis of the right part $P(x)$ structures of the ULD dictionary entries allows selecting the following structural elements. The largest structure is that which reproduces the effect of the register unit polysemy – let us call it *polysemy*. Thus, polysemy: C_r is a part of $P(X)$, which gives an interpretation of r -th meaning of the register unit X . Besides there are collocations in $P(X)$ – the word combinations of different types, which include the register unit X . They are marked with characters $F(i, j)$, where i is a collocation type, j is its number. Hence, an arbitrary $P(x)$ may be presented as unification:

$$P(X) = \cup_r C_r \cup \cup_{i,j} F(i, j) \quad (3.27)$$

In each rubric C_r there are illustrations of the r -th meaning, i.e. the examples of its use in the literary texts, which are marked with character $J(r, q)$, as well as dictionary entry fragments that reflect its shades of meaning. They are represented with two slashes «//»; let us mark a shade of the meaning C_r with $V(r, j)$. One or more illustrations may be given to each shade of meaning. Let us mark the k -th illustration of the shade of meaning $V(r, j)$ with $J'(r, j, k)$.

The collocations have the same structure. In the Ukrainian lexicography the collocations are traditionally presented at the end of the dictionary entry after the mark \diamond . In the existent dictionaries of the Ukrainian language the collocations are structurally divided mainly on two classes that are marked with \diamond and Δ . Such word combinations as complex pseudonyms are marked with \square . The terminological collocations are given after the mark Δ , all other kinds of collocations are given after the mark \diamond .

Since the basic principle of allocating the collocations is desemantization (partial or full) of their components, the collocations in the entry structure of the ULD-20 are not connected to the lexical meanings of these components. It's the main difference between the ULD-11 and the ULD-20. This approach allows formalizing a number of structural effects, abstracting from the rather complex, semantically ambiguous and theoretically still not developed problem of the correlation between the lexical and phraseological meanings.

But presenting the units different by their linguistic status, properties of functioning, internal and external features after the mark \diamond complicates their perception by the dictionary user. So an additional structuring of these dictionary elements is realized in the ULD-20 using a new element – word equivalents. Let us see the representation of the units of this class in the ULD structure.

The concept of *word equivalent* [2, 6] appeared in linguistics at the end of the last century. The word equivalents as language units that are separated from the word and from the idiom were first detected and ordered by R. Rogozhnikova in the “Dictionary of Word Equivalents”. She defines the word equivalents as “bound word combinations that are characterized by the stability, unity of the meanings, mainly invariable form” [6]. As word equivalents are the linguistic units indivisible at the syntactic level, they can be related to the class of syntaxemes of certain type.

These elements of the language system in the ULD-20 are singled out as register units in the word family of the main word beyond the mark \circ after the idioms.

Each collocation may have several meanings $FC(i, j, k)$. Several illustrations $J^{FC(i, j, k, m)}$ and shades of meaning $J^{FC(i, j, k, r)}$ are given to each meaning. And they in turn may have several illustrations $J^{FCV(i, j, k, r, l)}$.

Thus, index i runs over the following meanings in the system of collocations $F(i, j)$ of the ULD-20: <free word combinations>; <term word combinations>; <idioms>; <word equivalents>.

Taking into account the mentioned above, the following system of enclosures is fair:

$$\begin{array}{c}
 J(i, q) \\
 \cap \\
 C_i \supset V(i, r) \supset J^V(i, r, t). \\
 F(i, j) \supset FC(i, j, k) \supset J^{FC(i, j, k, m)} \\
 \cup \\
 J^{FC(i, j, k, r)} \\
 \cup \\
 J^{FCV(i, j, k, r, l)}
 \end{array} \quad (3.28)$$

Formulas (3.28) set a formal structure of the right part of the ULD dictionary entry.

The theory of lexicographic systems allows deepening the structure constructed by applying the procedure of the recursive reduction to its individual elements. This is achieved by the following way.

The main interpretation function is right entering each register word into the language system taking into account the peculiarities of its lexical semantics. The process indicated is based on the specifics of the internal structure of the lexical meaning and stipulates the existence of certain norms and methods for constructing the interpretation, governed by the rules of the lexicographic tradition.

Modelling the lexical semantics is a detection of the language means – words and connections between them, using which a decompression of the lexical meaning for the dictionary register words based on the interpretation formula is possible. The modelling is also an establishment of the groups, which are relative by different parameters.

The meaning of any linguistic unit appears in connection with other units. So it is very important to analyze the semantic phenomena in the context, i.e., to model their behavior in the real conditions of functioning.

In this case, for the interpretation formula components, the context is defined by this formula, which is syntactically one sentence – simple, extended, complicated with the homogeneous parts of the sentence. Such sentence as an interpretation formula organically fits into the entry and the dictionary as a whole. A definition is considered as a sufficient context for detecting the nature of its components. The interpretation formulas are standardized structures, within which the functioning of their components is definite and clear. So the interpretation formula in the structural and functional aspects is an autosemantic sentence not containing the indicators of the syntactic connections with other sentences and is sufficient for detection and definite qualification of its components. This is confirmed by the following statements:

1. The sentences, which are the full interpretation formulas, are the definite representatives of the semantic states for a lexeme.

2. The structure of sentences, which are the full interpretation formulas, contains everything necessary for the characteristics of the semantic state.

The recognition of these two statements leads to decomposition of the interpretation formula to the individual components – the determinants of meaning, which definition and characterization is performed by interpreting a set of interpretation formulas as a kind of lexicographic systems and applying the process of recursive reduction to it. This process will be performed according to the following positions:

1. The rule of decomposition of the interpretation formula into the components is one for all the verbs.

2. Each component has its generalized elementary sense and in a certain sense is a seme.

3. The main component of the interpretation formula, which is necessary and is a centre of the semantic state, is selected from the others. Let us call it a “semantic theme” (ST). There is one and only one ST for each semantic state.

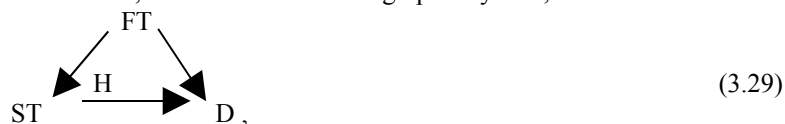
4. All other components are determined by the semantic theme and may be optional. Let us call them “differential semes” and mark a set of differential semes with D character.

5. The differential semes are in interaction with the semantic seme and with each other in a way that each of them reduces the semantic field size defined by the semantic theme.

6. A set of components of the verbal interpretation formulas is a universal linear structure.

7. The components of the interpretation formula are functional, but the unified forms for expressing a component are selected; paraphrasing the components of the interpretation formula in the direction of optimization and unification.

In this way, a set of interpretation formulas FT, considered as a lexicographic system, looks like:



where ST is a set of semantic theme, which plays a role of the register (formal) part of the lexicographic system FT, and a set of the differential semes D plays a role of the interpretation (substantial) part; H as always performs the connection between them. The formula (3.29) acquires a specific sense when putting a specific verbal lexeme x to it:

$$FT(x) = ST(x) \xrightarrow{H} D(x) = ST(x) + D(x) \quad (3.30)$$

By analyzing the dictionary definitions, the types of differential semes typical for the verbal interpretation formulas were selected and the typical structures of all components including ST were set. A set of 10 differential semes were detected and identified: Dsubj – „subject”; Dobj – „object”; Dadr – „addressee”; Dinstr – „method, means, material”; Dloc – „environment, place”; Dtemp – „time”; Daim – „goal”; Dcond – „condition”; Dcaus – „cause”; Dmsr – „extent, intensity”. The specified set is invariant; as a result any interpretation formula is decomposed into the linear (tabular) format with the following structure:

Semantic theme	Differential semes									
	subject	object	addressee	method, means, material	environment, place	time	goal	condition	cause	extent, intensity
1	2	3	4	5	6	7	8	9	10	11

Table 3. The format of verb definitions

The structure of the generalized interpretation formula for a verb looks as follows:

$$FT = ST + D = ST + D_{subj} + D_{obj} + D_{adr} + D_{instr} + D_{loc} + D_{temp} + D_{aim} + D_{cond} + D_{caus} + D_{msr} \quad (3.31)$$

The order of the components is not shown here (as well as a syntactic structure of the interpretation formula). The proposed format serves as an easier object than a dictionary definition, what makes it available for the automated research. The linguistic analysis of this format is carried out in [5] and the typical implementations of its individual components-differential semes in the ULD lexicographic system were detected. The systematization allowed conducting the lexical-semantic stratification of the verbal vocabulary. In particular it is shown that the unsubstantial semes represented by the semantic theme are the most abstract in the verb meaning. The most common unsubstantial semes, that determine the actual lexical meaning of the verbs as a part of speech, are the semes “action”, “state”, “relation”. At the lowest level of the hierarchy, the verb classes of action, state and relation are concretized concerning the nature of the action, state and relation by the specific semes, represented with a semantic theme in the interpretation formula structure. The classes are divided into the subclasses – lexical-semantic groups (verbs of motion, thinking, behavior, etc.). The direct appeal to analysis of the semantic themes shows that the unification of the analyzed verbs into the relevant lexical-semantic groups is provided by the most frequent ones among them, which relate to the concepts of movement, thinking, behavior, etc., respectively. For example, the ST “пересуватися” – “to move” is integral for the verbs *іти* (*їти*), *ходи ти*, *літа ти*, *леті ти*, *плавати*, *плисти* (*пливти*, *плинути*) etc.

In the work mentioned the ST for all verbal semantic states and, therefore, for all verbs are set. It allowed constructing a new type of dictionary – “Dictionary of Verbal Themes” – in the automated mode.

The method of tabular representation of the dictionary definitions has proved a reliable apparatus for establishing the correlative strata of the vocabulary by the differential semes that concretize the action concerning its subject, object, addressee, method, means or material, place or environment, goal, condition, cause, extent or intensity. Thus the specific semes peculiar to the individual lexical-semantic variations, as a rule, do not line up in a vertical line of dependency, but they are subordinated to the core seme presented by the semantic theme. A typology of the analyzed verbal lexemes is set in the research on

the basis of the object and adverbial elements of the semantic structure. In particular, the correlation of the verbal classes with the relevant subjective representing noun classes is automatically detected. As for the other differential semes, they are activated in the verb semantic structure when semantic themes, by which the action is transmitted, require the adverbial concretizers.

The analysis of the dictionary interpretations, converted into the tabular format, indicates that the basis of lexical-semantic system of the modern Ukrainian language is a number of universal semantic features, a combination of which forms an array of real lexical meanings of the verbs. The method of format representation of the dictionary definitions is a reliable apparatus for establishing the correlative sets of vocabulary by the differential semes. These semes peculiar to certain lexical-semantic variations usually are not subordinated to a certain hierarchy, but are in the relation of equality as linear elements of the format. Using this format at compiling the verbal entries allows unifying the interpretations of certain meanings, as well as promotes the fuller description of the semantics of certain lexemes, making the verb structure semantics more precise.

A similar theory is created for other parts of speech. Thus, based on the component analysis of the noun interpretations [3] on the material of the ULD LDB, for the noun lexical-semantic groups of names of the buildings and instruments, united by a common semantic seme of the functional purpose, it was found, that they belong to a structural-semantic interpretation type with the integral semantic component (seme) with the meaning of destination. Schematically the integral seme structure of this interpretation type may be presented as:

$$A_{01}^p = \{ST, T, A, L, F\}, \quad (3.31)$$

where, A_{01}^p is a noun interpretation formula; ST (semantic theme – semantic dominant) is a required component of the interpretation, that includes the reality to a higher conceptual level (generic feature); T (time) is an optional semantic component with a meaning of time; A (attribute) is an optional semantic component that means the external features peculiar to the realities; L (place) is an optional seme characterizing the spatial location of the reality; F (destination) is an optional semantic component that points to the destination of the reality, its use.

The semantic theme in the explanatory structures of the lexemes described is a required component because it matches the reality with the relevant generic concept and forms the semantic core of the definition. In most cases the interpretation is formed on the basis of a semantic theme, the formal implementation of which adds up to the noun form in the nominative case. In some interpretations several semantic themes function simultaneously.

Other parts of speech are not considered here, as forming the dictionary definitions for them does not significantly differ. But the goal of studying the underlying structure and constructing the relevant structures for the interpretation formulas of different parts of speech seems to be obvious. Such research is important in consideration of the task for automatic constructing the taxonomic classifications and diagrams that are implicitly put into the dictionary definitions. This in turn creates a bridge to defining the connections between the linguistic and ontological descriptions of the linguistic realities, as well as to implementing the more intellectual modes for using the explanatory dictionary. Construction and computer realization of the interpretation formula structures pave the way for further automation of the lexicographic work – to the possibility of the full automatic construction of the dictionary entries.

12. The Systems Engineering Basis for Constructing the ULD Lexicographic Database

Based on the theory developed and according to the project of compiling new 20-volume Dictionary of Ukrainian language the ULD linguistic database was created in ULIF. All the main elements of the L-system structure mentioned above are represented in the database. For this purpose the fundamental academic lexicographic system “Dictionary of the Ukrainian Language” was created in ULIF in the form of a computer tool set, which had a well-structured LDB of the explanatory dictionary and supported a number of functions for compiling the 20-volume dictionary.

To create the ULD LDB, the ULD-11 paper version was converted to the electronic form. This step was performed by scanning and recognizing the text. Then it was saved in the RTF format and printed for proofing in order to correct errors. After double-proofing and correcting the electronic text, it was

converted from RTF to HTML format with the Unicode encoding system using the text editor MS Word. It should be noted that the work with the LDB in the electronic text format of such a large volume (over 135 MB) is completely ineffective. In addition to a very slow pace of the system work with such text arrays, the fundamental disadvantage is the impossibility of direct access to certain dictionary elements. Thus the creation of the specialized ULD LDB is necessary. And its formation should be automatic, as creating such a large database in manual or semiautomatic mode is simply impossible.

To ensure the automatic conversion of the ULD text to the LDB, a special software for selecting its structure elements according to the L-system structure was developed in ULIF using the printing features of its textual identification. As a result of conversion, the entire text of the ULD was moved from the HTML-files to the LDB with the following structure.

The LDB structure is minimal. It allows proper displaying all the ULD structural elements, but also it has possibility for expanding them. For example, the left part of the entry in the LDB is not structured, and is recorded in a single block.

The ULD LDB structure and the client program allow not only presenting but also visualizing the presentation of any dictionary entry as a tree. This simplifies the access to the structural elements of the entry, and the connections between the elements become evident. Many auxiliary elements of the entry (terminal symbols, interpretations numbers, some punctuation marks, special characters and font selection) do not require saving in the LDB and can be dynamically added during the entry formation by the output program. Such automatic operations help to avoid many errors while editing the entries. So the possibility of erroneous entering the elements, which break the dictionary structure, is out of the question. The process of editing becomes more easy, controlled and unified. The operations of adding, removing and fixing the entry elements are easily performed.

A set of tables and connections between them described below, is singled out from the ULD LDB.

The table “Register Words of the ULD (“**nom**”)” has the following fields: **ID** – a unique entry identifier; **Reestr** – a register word of the ULD; **Part** – a part of speech code of the register word; **Data** – date and time when the latest correction was made; **Digit** – a register word digital code used for sorting (the Ukrainian alphabet letters in this code are marked with two digits: A – 01, Б – 02, В – 03 etc., the numbers are marked with four digits: 1 – 0001, 2 – 0002, ..., 10 – 0010 etc. All other characters are ignored); **IsLink** – an attribute, whether it is a reference entry (it is marked with “*дуб.*” in the entry); **LinkText** – a reference text; **IsOldSum** – an attribute, whether the entry belongs to the ULD-11 or whether it is new; **IsDel** – an attribute, whether the entry was removed from the DB; **QtyEd** – number of amendments for the entry; **FinalEd** – an attribute, whether the entry was coordinated with the main research editor; **NREd** – an attribute, whether the entry was coordinated with the research editor; **Printed** – an attribute, whether the entry was printed; **Odious** – an attribute, whether the entry belongs to problematic ones. The table is indexed by the following fields: **ID** (Unique), **Reestr**, **Part**, **Digit**.

The table “Editing Ranges of the ULD (“**Ranges**”)” has the following fields: **Part** – number of the technological volume (T-volume) or its part; **Lower** – a word which is the lower limit of the part; **Upper** – a word which is the upper limit of the part; **Letter** – the letter, which fully belongs to that part of the volume (if the field is not empty, then the values of the fields **Lower** and **Upper** are ignored and vice versa); **LexEd** – the name of the research editor or lexicographer responsible for T-volume or a part of T-volume.

The table “Left Parts of the Entries (“**lr**”)” has the following fields: **ID** – a unique entry identifier (it should match the relevant entry identifier from the table **nom**); **Left** – the left part text; **Right** – is reserved for the right part of the entry; **IsDel** – an attribute, whether the record is removed from the DB. The table is indexed by the field **ID** (Unique).

The table “Interpretation Blocks (“**intgroup**”)” has the following fields: **ID** – a unique block identifier; **ID_iv** – an identifier of the upper level fragment, with which the block is connected; **NumbGr** – number of block within the entry (the numbering must be consecutive); **IsDel** – an attribute, whether the record is removed from the DB; **Param** – a block parameter. The table is indexed by the following fields: **ID** (Unique), **ID_iv**, **NumbGr**.

The table “Interpretations (“**interpr**”)” has the following fields: **ID** – a unique interpretation identifier; **ID_iv** – an identifier of the upper level fragment, with which the interpretation is connected; **Relat** – a

code of the type for relation, which the combination with the register word expresses in this interpretation (only for the prepositional entries); **NumbInt** – number of interpretation within the fragment (the numbering must be consecutive); **IsDel** – an attribute, whether the record is removed from the DB; **Lv** – a code of the type for the upper level fragment. The table is indexed by the following fields: **ID** (Unique), **ID_iv**, **NumbInt**, **Lv**.

The table “Idioms and Word Equivalents (“**fraseol**”)” has the following fields: **ID** – a unique identifier of the idiom or the word equivalent; **ID_iv** – an identifier of the upper level fragment, with which the idiom or the word equivalent are connected; **NumbFras** – number of the idiom or the word equivalent within the fragment (the numbering must be consecutive); **Kind** – a type of the idiom or the word equivalent; **Fras** – a name of the idiom or the word equivalent; **IsDel** – an attribute, whether the record is removed from the DB; **Lv** – a code of the type for the upper level fragment. The table is indexed by the following fields: **ID** (Unique), **ID_iv**, **NumbFras**, **Lv**.

The table “Shades of Meanings (“**shade**”)” has the following fields: **ID** – a unique identifier of the shade of meaning; **ID_iv** – an identifier of the upper level fragment, with which the shade of meaning is connected; **NumbShade** – number of the shade of meaning within the fragment (the numbering must be consecutive); **Lv** – a code of the type for the upper level fragment; **IsDel** – an attribute, whether the record is removed from the DB. The table is indexed by the following fields: **ID** (Unique), **ID_iv**, **NumbShade**, **Lv**.

The table “Interpretation Parts or Idiom (Word Equivalent) Meanings (“**subshade**”)” has the following fields: **ID** – a unique identifier of the interpretation part or the meaning; **ID_iv** – an identifier of the upper level fragment, with which the interpretation part or the meaning are connected; **NumbSub** – number of the interpretation part or the meaning within the fragment (the numbering must be consecutive); **Lv** – a code of the type for the upper level fragment; **IsDel** – an attribute, whether the record is removed from the DB. The table is indexed by the following fields: **ID** (Unique), **ID_iv**, **NumbSub**, **Lv**.

The table “Interpretation Formulas (“**formula**”)” has the following fields: **ID** – a unique identifier of the interpretation formula; **ID_iv** – an identifier of the upper level fragment, with which the interpretation formula is connected; **NumForm** – number of the interpretation formula within the fragment; **Interpr** – the interpretation formula text; **Lv** – a code of the type for the upper level fragment; **Paradigm** – a paradigmatic class; **Vid** – a type; **Perexidn** – transitivity; **Keruvan** – government; **Spoluch** – combinative power; **Rid** – gender; **Chislo** – number; **Style** – style; **ElsOll** – other parameters of the interpretation formula; **IsDel** – an attribute, whether the record is removed from the DB. The table is indexed by the following fields: **ID** (Unique), **ID_iv**, **NumForm**, **Lv**.

The table “Illustrations (“**illustr**”)” has the following fields: **ID** – a unique identifier of the illustration; **NumbIll** – number of the illustration within the interpretation formula (the numbering must be consecutive); **Illustr** – the illustration text; **Author** – an author of the illustration source; **Title** – a name of the illustration source; **Edition** – a year of the illustration source publication; **Pages** – a page number from the illustration source; **Figur** – an attribute, whether the register word is used in the illustration figuratively; **Cm** – an attribute, whether the register word is used in the illustration in comparison; **IsDel** – an attribute, whether the record is removed from the DB; **ID_iv** – an identifier of the upper level fragment, with which the illustration is connected. The table is indexed by the following fields: **ID** (Unique), **ID_iv**, **NumbIll**.

Let us describe the connections between the ULD LDB tables. The LDB main table is **nom**, each record of it corresponds to one dictionary entry of the ULD. The **lr** table is connected with **nom** by the principle “one to one”, because each entry has only one left part. The connection is carried out by the **ID** field. The tables **intgroup**, **interpr**, **fraseol**, **shade**, **subshade**, **formula** and **illustr** include the fields **ID**, **ID_iv** and **Lv** (except for **intgroup** and **illustr** that have no **Lv** field). These fields are used for connecting the tables and their purpose in all of the tables is the same. The **ID** field is a unique identifier of the record; **ID_iv** is an identifier of the upper level, with which the record is connected; **Lv** determines what table (i.e. the entry fragment type) of the higher level is mentioned. For example, if **Lv** has a value of 0 in a certain record of the **interpr** table, then this record (interpretation) concerns the register word (it is not a part of the interpretation block), which ID equals the **ID_iv** value for this record.

Next, the tables that can be the immediate higher level for other tables are specified for each table: **intgroup** – **nom**; **interpr** – **nom**, **intgroup**; **fraseol** – **nom**, **interpr**; **shade** – **interpr**, **fraseol**, **subshade** (only

when it relates to fraseol); **subshade** – interpr, fraseol, shade (only when it relates to interpr); **formula** – interpr, fraseol, shade, subshade; **illustr** – formula.

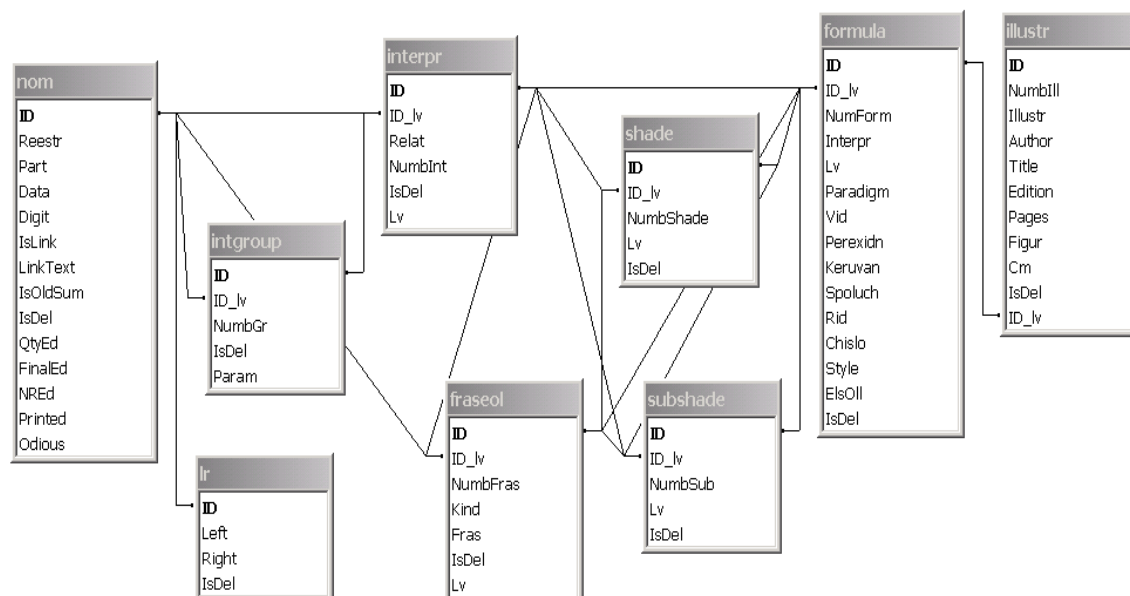


Fig. 1. A chart of connections between the ULD LDB tables

As one can see from this list, the tables connected with only one table does not have the Lv fields.

The meanings that Lv acquires depending on the table of the higher level: 0 – nom; 1 – intgroup; 2 – interpr; 3 – fraseol; 4 – shade; 5 – subshade.

The connections between the ULD LDB tables can be interpreted in accordance with the ULD structure in the following way:

1. The ULD dictionary entry always has the left part.
2. An entry may consist of several (1 or more) interpretation blocks, interpretations, idioms or word equivalents (if the word is used only as a part of the idiom or the word equivalent).
3. A block can consist of several interpretations.
4. An interpretation can have the shades of meanings, parts of interpretation and the idioms and the word equivalents related with it.
5. An idiom and a word equivalent can have several meanings and shades of meanings.
6. A meaning of the idiom or the word equivalent can also have the shades of meanings.
7. An interpretation, a shade of meaning, a part of interpretation, an idiom (a word equivalent), an idiom (a word equivalent) meaning include the interpretation formula (sometimes they may not include it, then we consider that there is a dummy (empty) interpretation formula) and may include the illustration.
8. An interpretation formula is actually a component of the interpretation, shade of meaning, part of interpretation, idiom (word equivalent), idiom (word equivalent) meaning. But the structures of these interpretation formulas are similar, so they are presented in the **formula** table. The table stores the interpretation formula text and its grammatical, stylistic and other parameters. Certain fields are reserved for these parameters, but today all the parameters are stored in the **ElsOll** field.
9. The text illustrations with the examples of using the register word in certain meanings may be available only when there is an interpretation formula for the relevant meaning. So the illustration table **illustr** is connected only with the interpretation formula table **formula**. There are instances when there is actually no interpretation formula, but only the parameters – then a record with a zero value of the **Interpr** field is created in the formula table. An idiom and a word equivalent also may not have the interpretation formula - when they are references or have several meanings. In both these cases, the relevant record in the **formula** table may not be created, so such an idiom and a word equivalent has no own illustrations. The structures of the **intgroup**, **interpr**, **fraseol**, **shade** and **subshade** tables are very similar, but they have several differences:

- **intgroup**: a block has no interpretation formula, but may have parameters. Therefore, its parameters are a direct part of the **intgroup** table and are recorded into the **Param** field;
- **interpr**: an interpretation may be characterized by the relation, the code of which is written into the **Relat** field. For interpretations with the same relation type, which belong to one higher level and have consecutive numbers, the type is specified only once for the first interpretation of the interpretation group while forming the entry;
- **fraseol**: an idiom always has the name (i.e. the idiom text, which necessarily includes the register word), which is recorded into the **Fras** field. An idiom has the **Kind** field for the idiom type marked with a number from 1 to 5. The type determines how an idiom will be marked in the entry. The idioms of the same type, which belong to one higher level and have consecutive numbers, are combined in a block, which mark appears only once;
- **subshade**: this table has no distinctions in the structure, but its peculiarity is that it stores the interpretation parts and shades of meanings, as well as the idiom meanings. It is considered that when an idiom (or its meaning) has a shade, than this shade can not have parts, otherwise it would have led to appearing a cycle in the structure of table connections.

The **illustr** table has the **Illustr** field with the direct illustration text, and the **Author** field for storing the author's name or the illustration source name. The **Title**, **Edition** and **Pages** fields are created only for the compatibility with the ULD-11 and are not used in the ULD-20, but if they are filled in, they appear in such sequence while forming the entry (about the **Figur** and **Cm** fields see the description of the **illustr** table).

The tables **intgroup**, **interpr**, **fraseol**, **shade**, **subshade**, **formula** and **illustr** have the fields for the internal consecutive numbering of the records within the entry or its fragment. These fields are the following for the tables: **intgroup** – NumbGr; **interpr** – NumbInt; **fraseol** – NumbFras; **shade** – NumbShade; **subshade** – NumbSub; **formula** – NumForm; **illustr** – NumbIll.

The interpretation formula is actually always only one, that is why the value of the NumForm field should be always equal to 1.

The ULD LDB implemented in ULIF functions under the DBMS Microsoft SQL Server 8.0. The client program of editing the ULD LDB was designed and created in the Microsoft Visual Studio 6.0 environment. It works under the Microsoft Windows 2000, Microsoft Windows XP or Microsoft Windows Vista operating systems.

The program is oriented for work in the network environment where a lot of users simultaneously have access to the ULD LDB. In this case, depending on preference, the users can obtain the access to the entire database or its part, the possibility for editing the entries or just viewing them.

The client program implements a lot of functions to work with the LDB. These functions are performed by separate modules of the program.

DicUASplApp is the main module of the program. It includes the functions for work with a dictionary entry as a whole: adding, removing, copying, moving to the editing of an entry, setting the attributes of editing, writing the entries to a file for the following printing, as well as the functions associated with reviewing the entries: setting a font, selecting a filter mode (by part of speech, editing range, arbitrary request). In addition, the module includes a number of system functions: checking the belonging of a word to the desired range, converting a word into the digital code, the reaction to pressing certain keys etc. The main global variables of the program are also described here.

DicUASplView is the main window of the program. It includes the functions of selecting an editing range and a dictionary entry for viewing, of finding a register word, as well as system functions for initializing the ULD database and the tree of editing ranges.

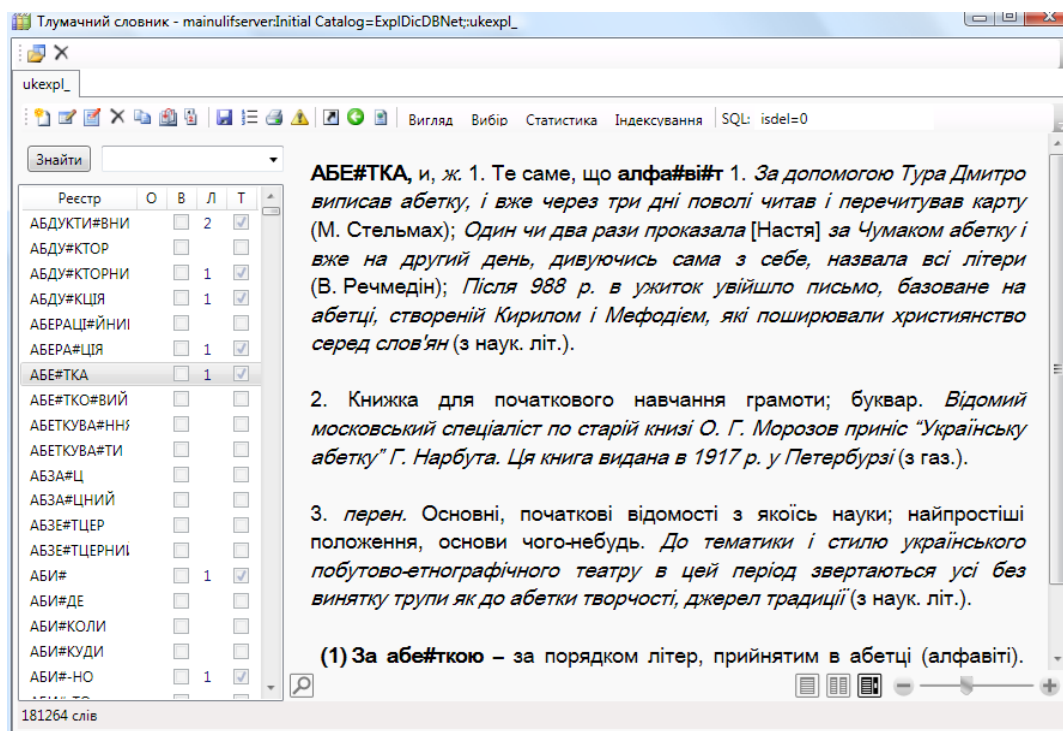


Fig. 2. The main window of the ULD program

ArtTree – viewing the entry structure. The functions of adding and removing separate entry elements, editing them, reordering and setting the attributes of editing can be called from here. If the program is called only in the view mode, you can navigate the entry structure, but the editing functions will not be available.

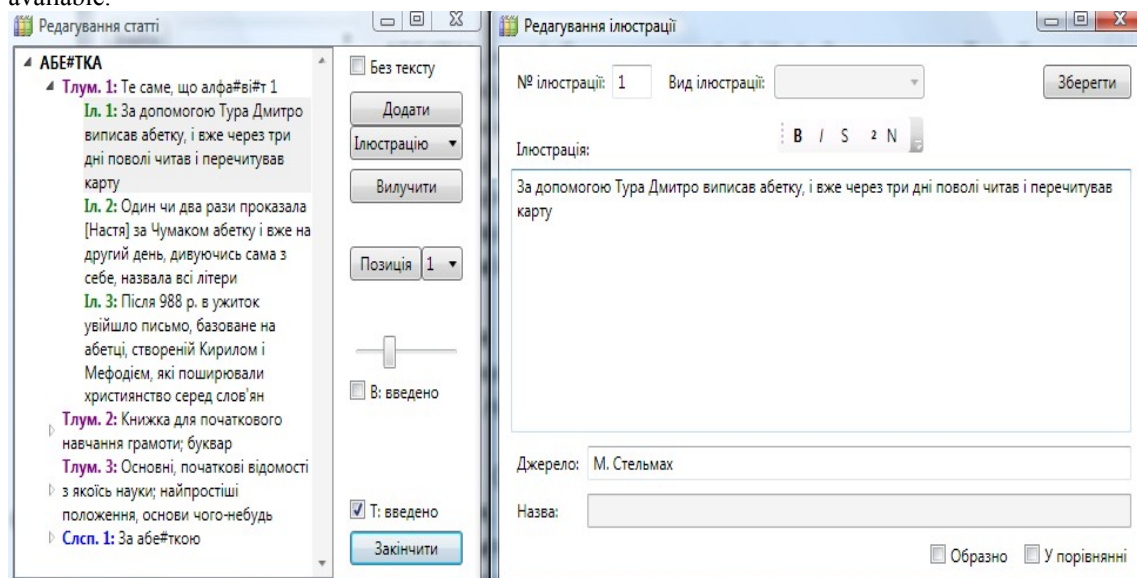


Fig. 3. The mode for viewing structure and editing dictionary entries

At the end of this section let us note that a tool set for compiling the ULD-20 was performed with the systems engineering of the virtual lexicographic laboratories that were described in the proceedings of MONDILEX Second Open Workshop [4].

Bibliography

- [1] *Dictionary of the Ukrainian Language*. (1970-1980): Dovira, Kiev, in 11 volumes. (in Ukrainian)
- [2] Luchik, A. (2001). *Semantics of the Adverbial Word Equivalents of the Ukrainian and Russian Languages*: Dovira, Kiev, 218 p. (in Ukrainian)
- [3] Pohribna, O., Chumak, V., Shyrovkov, V., Shevchenko, I. (2004). Linguistic Classification of the Ukrainian Noun in Light of the Theory of Lexicographic Systems. In: *Movoznavstvo*, vol. 5-6, pp. 62-82, Kiev. (in Ukrainian)
- [4] Rabulets, A. (2009). Systems Engineering Principles of Virtual Linguistic Laboratories. In: *Proceedings of MONDILEX Second Open Workshop*: Kiev, 2-4 February, pp. 18-23.
- [5] Rabulets, A., Sukharyna, N., Shyrovkov, V., Yakymenko, K. (2004). *The Verb in the Lexicographic System*: Dovira, Kiev, 260 p. (in Ukrainian)
- [6] Rogozhnikova, R. (1991). *Dictionary of Word Equivalents: Adverbial, Auxiliary, Modal Unions*: Moscow, 254p. (in Russian)
- [7] Shyrovkov, V. (2005). *Elements of the Lexicography*: Dovira, Kiev. (in Ukrainian)

Using Ukrainian National Linguistic Corpus in Lexicography

Oleg Bugakov

Ukrainian Lingua-Information Fund, National Academy of Sciences of Ukraine

Abstract. The search parameters of the Ukrainian National Linguistic Corpus are described. The dictionaries created using the Ukrainian National Linguistic Corpus in the Ukrainian Lingua-Information Fund are mentioned. The ways of using the Corpus for creating these dictionaries are specified. The structure of the semantic dictionary of prepositional phrases is described.

Keywords: Ukrainian National Linguistic Corpus, explanatory dictionary, Ukrainian Language Dictionary, semantic dictionary of prepositional phrases, dictionary of synonyms.

The Ukrainian National Linguistic Corpus

A number of dictionaries are now created in the Ukrainian Lingua-Information Fund, NAS of Ukraine (ULIF), on the basis of the Ukrainian National Linguistic Corpus (UNLC). Corpus volume is about 54 million words. The corpus is presented by the texts of different styles and genres without proportions. If necessary the researcher can create subcorpora for different styles according to the statistical parameters.

There are two types of search in UNLC: by bibliographic attributes and full text search using modern linguistic technologies [6]. The search by the bibliographic description is intended primarily for selecting a subarray of information for further processing.

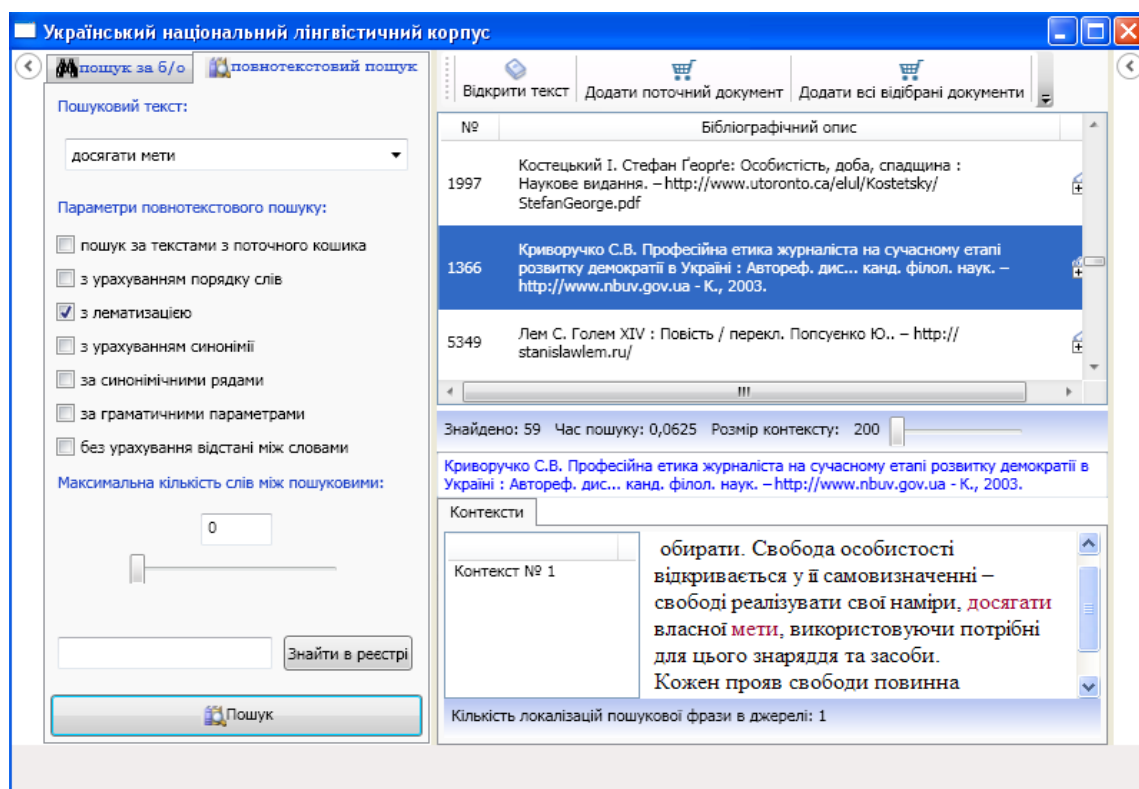


Fig. 1. The Ukrainian National Linguistic Corpus

The full-text search is carried out after the previous procedure of indexing the texts in the UNICODE, confronted with the objects of storage of the electronic library. For the full-text search you must enter a search phrase and set the full text search parameters.

The full-text search can be executed with the following parameters:

- taking into account the word order;
- with lemmatization;
- taking into account the synonymy;
- by synonymic rows;
- by grammatical parameters;
- without taking into account the distance between words;
- semantic search.

After the full text search the user can view the contexts of search phrases in the selected text. When you select one of the objects in the search results, the search of the contexts takes place within the text indexed.

The search words of the context in the text are highlighted with a certain colour, for example, in the localization of the search phrase *досягнути мети* (*to reach the goal*) the word forms *досягнути* and *мети* are highlighted in red that correspond to the search phrase when searching with lemmatization (Fig. 1).

Creating the Explanatory Dictionary

A 20-volume explanatory Ukrainian Language Dictionary (ULD) as well as its electronic version is now created in ULIF. At the moment the registry of the dictionary is more than 181 thousand words. We plan to increase it up to 200 thousands (Fig. 2).

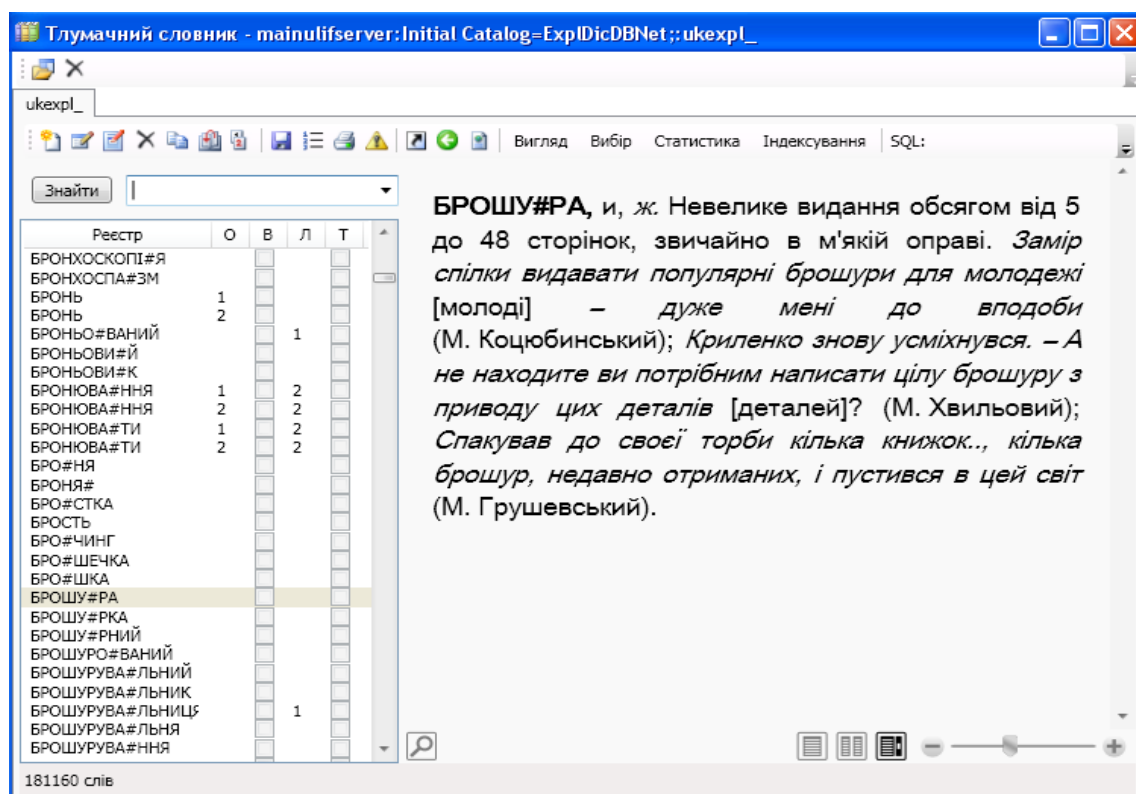


Fig. 2. The explanatory Ukrainian Language Dictionary

Using the text corpus we find new words, i.e. increase the dictionary registry. In addition, using contexts we find new meanings and shades of the meanings for the words that are already in the dictionary.

UNLC is the main source of illustrations for word meanings. Broad search capabilities help to choose the most accurate illustrations. We can search for the word «as is», i.e. in the grammatical form, what we need; search for the word / phrase in all grammatical forms; search for the word / phrase only in the specified grammatical forms, as well as search by mask (search by any part of the word).

Creating the Dictionary of Synonyms

The electronic dictionary of synonyms of the Ukrainian language is formed on the basis of 2-volume dictionary of synonyms. The dictionary is built as an information system in which a specific meaning (or shade of meaning) from the explanatory dictionary is presented explicitly for each element of the synonym set. The meanings, by which these lexical-semantic variants are combined into synset, are presented explicitly. This common meaning may be considered as its semantic invariant. The hierarchical order of semantic distance from the common semantic invariant is set between the elements of the synset [3].

UNLC is used as a material for creating the dictionary of synonyms. The lexicographers have the opportunity to view the contexts with the words that belong to several synsets.

Creating the Semantic Dictionary of Prepositional Phrases

The software of UNLC allows creating the specialized subcorpora oriented on solving the specified tasks. Using a specially designed program in ULIF, the subcorpora are converted into the databases with a certain structure that focuses on specific linguistic researches. The linguistic databases (LDB) function as tools and material for studying the linguistic phenomenon. They are structured on the following principle: the text segments (contexts) that contain a specific language unit, correspond to the predetermined differential features that are the basis for performing the analysis. Structuring LDB by the fields that meet the set of parameters for the analysis of diagnosing contexts, and the organization of access to these fields allow to automatically classify the material for each of the parameters and any combination of them.

The linguistic database of prepositional syntactic connections zone with a volume of 20.768 contexts was a basis for creating the semantic dictionary. The texts from the newspapers and magazines chosen from UNLC were the initial text material for forming the database. The general volume of these texts in corpus is 6 million tokens [2].

By means of special program the concordances were formed for each preposition on the morphologically marked texts. The limits of contexts were defined with beginning and end of the sentence where certain preposition is located. The database was structured by the following fields: (1) "Context", (2) "Length of prepositional syntactic connections zone", (3) "First position of preposition", (4) "Main word postposition", (5) "Contact main word", (6) "Main word", (7) "Main word code", (8) „Main word semantic class", (9) "Contact dependent word", (10) "Dependent word", (11) "Dependent word code", (12) „Dependent word semantic class", (13) „Relation", (14) „Notes".

The prepositional syntactic connections zone includes a preposition, main word (the word that manages a prepositional phrase) and dependent word (the word submitted to the main word by means of the preposition). At the semantic level the prepositional syntactic connections zones are considered from the point of view of the semantic interpretation of syntactic connection between the main word (MW) and dependent word (DW) [2].

The theoretical prerequisites for creating the dictionary is an investigation of preposition semantics in the formalism of the theory of semantic states [4, 5], because the prepositional constructions are the object of lexicographing, and their semantic states are the elements of the interpretative part.

According to this theory any word (unit of language) in the context or in linguistic flow is in certain semantic state that is a sum of grammatical and lexical semantics features for the lexical level units [4].

The semantic state of a preposition is a realization of the concrete semantic relation in the text between main and dependent words, caused by the semantic states of the latter [1].

Conducting investigation for identifying the set of typical semantic states for the prepositions is a necessary precondition for creating the specified dictionary. Defining an aggregate of semantic states that prepositions with the syntactically connected words express taking into account the semantic attributions of MW and DW, precedes establishing the semantic states of prepositions.

As a result of the analysis led on the material of the linguistic database we picked out 20 types of semantic relations that prepositions may express in the text.

In the text each type of semantic relations realizes as a rule a certain multitude of concrete relations depending on the semantic state of the preposition. The semantic state of preposition is defined according to the interpretation of the theory of semantic states of language units with six parameters: the relative semantics of preposition (quasilexical meaning), the case with which the preposition manages the dependent word (quasigrammatical meaning), and the lexical and grammatical semantics of the main and dependent words. Taking into consideration these parameters we picked out 131 semantic relations within analyzed database.

The obtained system of semantic relations underlay a semantic dictionary of prepositional phrases (Fig. 3). An interpretation of the concrete prepositional combinations while creating the dictionary was derived by the confrontation of grammatical and semantic information of the main and dependent words and the potential possibilities of preposition to express semantic relations.

A scheme of dictionary entry may be depicted as:

MW pos. PREPOSITION DW pos. in ...case || DW semantic class; MW semantic class; semantic relation.

MW is given in the initial form and DW – in the form of the case that preposition requires. A component that corresponds to the grammatical meaning of the preposition, that is a case of the dependent noun, is given in the block of DW grammatical meaning. To increase the visual effect, the left part is separated from the right one by the two straight lines. Only the titles of the semantic relations are given, their interpretations are given in the separate file. Such form of presentation will facilitate the automatic search for prepositional constructions by the parameter of semantic relation.

Let's see an example of the dictionary entry:

СТОЯТИ дієсл. **БІЛЯ НАБЕРЕЖНОЇ** ім. у род. в. || перебування; простір; Просторове 19.

There are three structural components of the left part:

k_1 – **СТОЯТИ** (MW), k_2 – **БІЛЯ** (preposition), k_3 – **НАБЕРЕЖНОЇ** (DW).

There are six structural components of the right part:

$G(k_1)$ – дієсл. “verb” (grammatical component of MW semantic state), $G(k_2)$ – род.в. “genitive” (quasigrammatical component of preposition semantic state), $G(k_3)$ – ім. “noun” (grammatical component of DW semantic state).

$L(k_1)$ – перебування “stay” (lexical component of MW semantic state), $L(k_3)$ – простір “space” (lexical component of DW semantic state), $L(k_2)$ – Просторове 19 “Spatial 19” (quasilexical component of preposition semantic state).

Structuring of dictionary entry of the created dictionary provides eventuality for information retrieval by every structural component: by preposition, MW and DW, by their grammatical characteristics, that is by part of speech, by DW case, by semantic classes of main and dependent words and by semantic relations. Information retrieval may be performed by separate parameters or by their aggregate.

The dictionary is an open system that may continually widen with new prepositional phrases. Now there are more than 13,000 prepositional phrases in it.

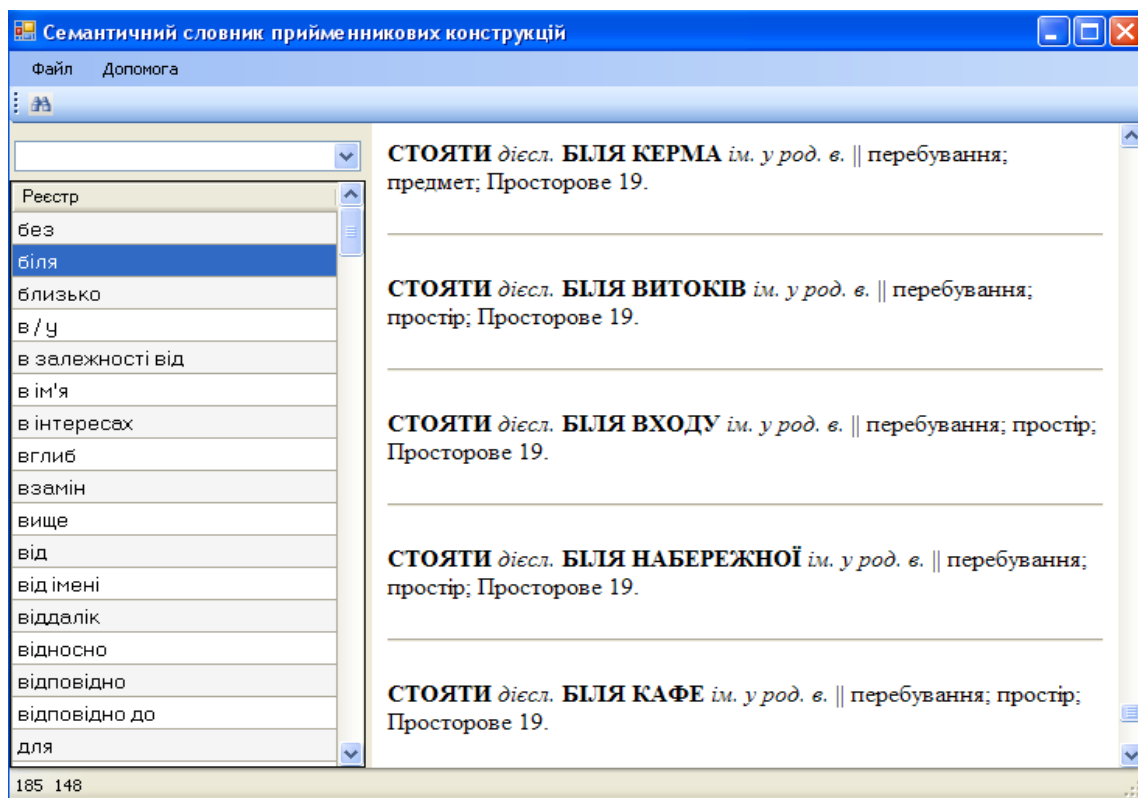


Fig. 3. The semantic dictionary of prepositional phrases

The dictionary may be used for further researches, in particular, for study of synonymic and antonymic prepositions. Besides it may be used in systems of natural language processing, particularly in linguistic analyzers as a source of lexical information while identifying prepositional phrases in text.

The UNLC can also be used when creating a spelling dictionary as a source of replenishing its registry; for the terminological dictionaries as a source of replenishing its registry and search for illustrations, and also for creating other types of dictionaries.

Bibliography

- [1] Bugakov, O. (2008). Semantic States of Ukrainian Prepositions. In: *Études Cognitives*, vol. 8, Warszawa.
- [2] Bugakov, O. (2005). Prepositional Syntactic Connections Zones in the Syntactic Structure of the Ukrainian Language. In: *Movoznavstvo*, vol. 5, pp. 75-87, Kiev. (in Ukrainian)
- [3] Griaznukhina, T., Ustimets, Y., Shyrovkov, V. (2008). Operational Defining the Semantic Closeness of the Synonyms in the Limits of Synset. In: *International Scientific Conference MegaLing'2008. Horizons of Applied Linguistics and Linguistic Technologies: DIP*, Simferopol, pp. 207-208. (in Russian)
- [4] Shyrovkov, V. (2005). *Elements of the Lexicography*: Dovira, Kiev. (in Ukrainian)
- [5] Shyrovkov, V. (2005). Semantic States of the Language Units and Their Use in the Cognitive Lexicography. In: *Movoznavstvo*, vol. 3-4, pp. 47-62, Kiev. (in Ukrainian)
- [6] Shyrovkov, V., Bugakov, O., Griaznukhina, T., etc. (2005). *Corpus Linguistics*: Dovira, Kiev (in Ukrainian)

Authors

Oleg Bugakov, Ukrainian Lingua-Information Fund, National Academy of Sciences of Ukraine, Kiev, Ukraine.

Ivan Derzhanski, Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria.

Ludmila Dimitrova, Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Sofia, Bulgaria.

Peter Ďurčo, University of SS. Cyril and Methodius in Trnava, Trnava, Slovakia.

Tomaž Erjavec, Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia.

Darja Fišer, Dept. of Translation, Faculty of Arts, University of Ljubljana, Ljubljana, Slovenia.

Radovan Garabík, E. Štúr Institute of Linguistics, Slovak Academy of Sciences, Bratislava, Slovakia.

Kristina Hmeljak Sangawa, Department for Asian and African Studies, Faculty of Arts, University of Ljubljana, Ljubljana, Slovenia.

Leonid Iomdin, A.A.Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia.

Jan Jona Javoršek, Department of Experimental Particle Physics, Jožef Stefan Institute, Ljubljana, Slovenia.

Violetta Koseska, Institute of Slavic Studies, Polish Academy of Sciences, Warsaw, Poland.

Natalia Kotsyba, Institute of Interdisciplinary Studies, Warsaw University, Warsaw, Poland.

Antoni Mazurkiewicz, Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland.

Adam Radziszewski, Institute of Informatics, Wrocław University of Technology, Wrocław, Poland.

Danuta Roszko, Institute of Slavic Studies, Polish Academy of Sciences, Warsaw, Poland.

Roman Roszko, Institute of Slavic Studies, Polish Academy of Sciences, Warsaw, Poland.

Volodymyr Shyrov, Ukrainian Lingua-Information Fund, National Academy of Sciences of Ukraine, Kiev, Ukraine.

Victor Sizov, A.A.Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia.

Petra Vide Ogrin, Slovenian Academy of Sciences and Arts, Library, Ljubljana, Slovenia.

Špela Vintar, Dept. of Translation, Faculty of Arts, University of Ljubljana, Ljubljana, Slovenia.