Tools for Generation of Natural Inflected Language Processors

Nadiya Mishchenko[+], Anatoliy Doroshenko[++]

National Academy of Sciences of Ukraine
03187, Kiev, Ukraine

[+]Institute of Cybernetics, Glushkov prosp. 40/1, nady@dolphin.icyb.kiev.ua
[++]Institute of Software Systems, Glushkov prosp. 40/5, dor@isofts.kiev.ua

Abstract. Supporting multiple languages and natural language processing are of high importance in information systems. This paper discusses software tools for the generation of languages processors (LPs) for the natural inflected languages. The tools are implemented in the LP generator DUAL, which allows for formal specification and reusability of developed components. The declarative language Dual is used to specify words, idioms, and their processing. The paper describes the automatic generation of dictionaries from their specifications in the Dual language and the reusability of software components, which facilitates fast construction of user-oriented software systems for processing of natural inflected languages. The LPs generated are intended for word-for-word translation of domain-specific texts in inflected languages and the generation of frequency lists of words and phrases used in statistical analysis of texts in inflected and analytical languages using Cyrillic or Latin alphabets.

1. Introduction

The development of natural language processors is of high importance in many applications. The applications may be usual, such as processing texts published on the Internet or searching text databases; or advanced, such as supporting knowledge bases with natural language interface or processing natural language in large software systems engineering etc. The main task in developing the natural language subsystems is the generation of language processors (LPs) for the natural languages. In information systems the task of LPs generation to provide linguistic support is more complex due to the need to take into account significant (semantic) information of the natural language texts. In fact, semantic information is an invariant of the text presented in different languages. Extraction of this information from the text is dependent on the language and can be facilitated by the tools that perform morphological and syntactical analysis.

The active phase of the LPs development for the Ukrainian language began in several institutions after it was announced that Ukrainian is the official language of the sovereign state Ukraine. In recent years our research group have developed a family of LPs for processing scientific and technical texts in natural inflected languages, particularly in Ukrainian and Russian. Our approach to the development of LPs is based on the formal methods we used to generate compilers for formal languages (declarative and procedural) [Mi98]. This approach facilitates the development of processors for the natural languages and allows us to study the effectiveness of these methods in a new domain.

We found that the LPs to be developed have some common functions and data. For example, they should perform morphological analysis and search for agreement among words. Moreover, the LP can process the texts in different inflected languages when the morphology of these languages is presented in the unified computer data structures designed for this purpose. So, the reusability of software components and the unification of data structures were used in software tools [Mi00] implemented in the natural LP generator DUAL that has some similarities with compiler generators.

In general, the main idea of implementing an LP generator consists in clearly separating the tools into two categories: those responsible for describing the task ("What should be done?") and those responsible for completing the task ("How to do it?"). Task description given by the user is a declarative specification that becomes an input to a generator, which converts it into data structures. The second question is answered by the text processing software tools implemented by the generator developers using universal software components of generator. Developed for one LP, these components can be reused later when a new LP is generated. The reusability of software components makes LPs more reliable, simplifies debugging and makes it possible to validate some requirements for the LPs at the specification level.

Generators were applied successfully for compiler generation for formal languages (declarative and procedural) [Ba97, Kr92], adaptable software and retargetability [Do02, Go02]. A compiler generator adopts a class of input languages defined by certain formal grammar and generates a set of compilers with common syntactical analysis. It is known by experience that a generator as a tool for LP generation is justified even when it is used to generate only one LP.

The LP generators for formal and natural languages have common architecture (but its parts have different contents) and the same strategy of the LP generation. The essential difference between them lies in different levels of formalization of the processes performed by the LPs. Due to the highly developed theory of formal grammars and compilation there are many languages for formal language specification. Similarly, development and application of the LP generator for natural languages are possible when formal specification of text processing is used. To this end, the special user-oriented declarative language Dual was proposed for morphology specifications, and software tools implemented in DUAL generator responsible for converting specifications into unified computer data structures were developed [Mi00].

This paper describes the LP generator DUAL approach and its applications. Section 2 presents informal list of requirements for generator DUAL development and its architecture. Section 3 outlines DUAL input specifications. Section 4 discusses applications of the LPs.

2. Requirements for the DUAL generator and its architecture

The development of DUAL, a system for the generation of LPs for the natural languages, is based on the following principles:

(1) DUAL is intended for word-by-word processing of scientific and technical texts in the inflected languages, in particular, Ukrainian and Russian. The scope of applications of DUAL allows for the efficient implementation due to following features:

- word-by-word processing is easy to specify and implement;
- the language of the specialized texts has limited vocabulary;
- the specialized texts are expected to have no homonyms;
- the inflected languages allow for an efficient detection of phrases (sequences of agreed words) using the results of the morphological analysis of the text. This makes an elaborated syntactic analysis unnecessary.

(2) The development of the LPs is based on the assumption that a dictionary for the LP includes words of three types: auxiliary words (prepositions, conjunctions etc.), common words and words from the vocabulary of some field of knowledge. Accordingly, separate word specifications should be developed. The dictionary compiled can include words of any type, depending on the intended task of the LP.

(3) The lexical units for specification are selected from the professional texts of the user. The resulting dictionary includes the up-to-date terminology.

(4) DUAL must include tools to support the development of specifications and the verification of specifications developed.

The architecture of DUAL includes six components*: {T,* Dual*, sflex, slex, D, Gen}*. Here *T* is a system of universal software components that process the professional texts; they perform morphological analysis, recognize phrases, form frequency lists of words or phrases etc. Dual is a declarative specification (meta) language, *sflex* is a specification of inflected language endings in Dual language; *slex* is a specification of lexemes for statistical analysis or translation in Dual language. The specification is a text divided into modules according to the types of lexemes: auxiliary, common or professional ones. *D* is a dictionary (computer representation of *slex* specification). At the beginning of LP generation, *D* usually contains auxiliary words of the input language. Words of other types are added during text processing. *Gen* is the generator itself.

Gen consists of three LPs: MORF is the generator of computer presentation of endings *mflex* from *sflex* specification; CON is the generator of dictionaries (monolingual or bilingual dependently on the task of the LP generated) from *slex* specification; GENW is a generator of all forms of words from specifications *slex* and *mflex* to verify them both. All generators listed have been implemented with the compiler generator "Terem" [Mi98] because Dual is a context-free language which can be used by "Terem" as input.

To conclude the section we note that parts *T* and *Gen* of DUAL are fixed components that the user cannot modify, while *sflex, slex* and *D* vary depending on an input language.

3. Input specifications of DUAL

In this section, three types of specifications are considered: the specification of endings, the bilingual specification of translation, and the monolingual specification of lexemes used for the statistical analysis.

(1) Specification of endings

From the point of view of morphological analysis implemented in the generator DUAL, every form of a word consists of at most three parts: stem (obligatory part common for all forms of the word), suffix, and ending. Ending is used in the traditional sense with some exceptions adopted in the computer linguistics. For example, absence of ending is treated as a presence of zero ending.

Suffix is a part of the word between stem and ending. If suffix is common for all forms of the word, then it may be joined to stem. Suffix in DUAL is allowed to consist of several canonical grammatical suffixes or may not coincide with any of them. Suffixes are described in lexeme specifications.

The biggest part of specification of endings consists of corteges of endings. Each cortege is represented by a finite sequence of endings typical for a certain class of inflectional words. In fact, cortege defines a class of words. Every cortege has mnemonic name. The name is formed from symbols of alphabet of the language specified. The first symbols should identify the grammatical categories of a word. For example, the name beginning with "nf" means noun, feminine gender.

Position in a cortege corresponds to the case for declinable words and to the person for conjugated words. Both, cases and persons have names. Symbol '0' in a cortege position identifies zero ending in this case (person). Symbol '.' in some position shows that the word cannot have ending in this position. Total number of positions is common to all languages and equal to 14.

Inflectionless words form classes with names of parts of speech words belong to: preposition, adverbs etc. It is said that inflectionless word coincides with its stem.
The final specification of endings consists of: the small and capital letters of input language alphabet, sequence of names of cases and persons, the sequence of cortege names, the sequence of corteges. From this specification LP MORF builds several representations, in particular, the list of endings and the machine representation of endings and corteges.

The next sections show how to use corteges, cases and its names in other specifications.

(2) Specification of translation

Specification for the word-for-word translation (bilingual specification) between inflected languages in language Dual is a sequence of rules ending with symbol ';'. Each rule begins with input sequence of symbols (inflectionless word, or stem common for several forms of inflectional word, or idioms, or complex preposition or complex conjunction to be translated) followed by an appropriate sequence of symbols of output language. Then, if necessary, one or more so called schemes of translation complete the rule. Scheme of translation presents grammatical information (names of ending corteges, names of positions in corteges, suffixes). This information is used for analysis of input words and for synthesis of output words.

In general, the syntax of rule of bilingual specification looks like following.

```
        <w1> => <w2>[<action>][: <scheme>{,<scheme>}
           {! <w2> [<action>][: <scheme>{,<scheme>}]}];
```

<w1> denotes the input sequence, <w2> denotes the output sequence. They are separated by the symbols '=>'. If the input sequence coincides letter-by-letter with the output one, then the symbol '*' is used instead of the output sequence. Square brackets denote optional entries of elements in brackets; braces denote repetition of the elements in braces zero or more times. <action> denotes the name of individual action used in the translation of w1.

Content of <scheme> will be explained by following simple examples of translation rules from Russian into Ukrainian. English equivalents for words in Russian and Ukrainian are given in the explanations followed the rules. To make the rules more understandable, the names of ending corteges and cases are given in English.

If an inflectionless word, or an idiom, or a complex part of speech is translated, then the scheme of translation is absent, as can be seen in following example:

```
часто => * ;
```

Russian word 'часто' (for English is 'often') coincides with Ukrainian word, so symbol '*' is used instead of output word.

```
прежде всего => передусім ;
```

Complex adverb 'прежде всего' (for English is 'first of all') is translated from Russian into Ukrainian adverb 'передусім'.

If an inflectional word is translated, then the scheme of translation is necessary. In bilingual specification, it always consists of two parts separated by the symbol '=', for example:

```
множеств => множин : nno = nfa ;
```

Here the input sequence is the stem of Russian word ‘множество’ (for English is noun ‘set’). This Russian word has suffix -ств- attached to the stem because declination does not change the suffix. The output sequence is the stem of the Ukrainian word with suffix -ин- also attached to stem. Left part of the scheme consists of name nno of cortege of endings taken by all nouns of neuter gender with ending -o in the nominative case. In right part of the scheme, there is the name nfa of cortege of endings of nouns, the feminine gender, with ending -a in the nominative case.

(3) Lexeme specification for text statistical analysis

Statistical analysis of texts is usually based on frequency lists formed using the results of the morphological analysis of words. The monolingual specification of lexemes differs from the bilingual one in two aspects.

Firstly, in the specification rule, instead of the output sequence, the optional information can be placed at user's discretion. For example, it may be the name of the field of knowledge the lexeme belongs to, or some abstract notion, which identifies the class of words containing this word, or any other textual information, or symbol '*' which is used to identify auxiliary words or idioms.

Secondly, the scheme consists of the grammatical information for only morphological analysis of input words. Below, we give the examples of Russian words specification obtained from bilingual specifications presented earlier.

```
часто => * ;
прежде всего => * ;
множеств => math :nno;
```

As another example let us consider monolingual specifications of words in German. Due to the inflexibility of this language, it is easy to use its morphology in the unified data structures shown for Russian and Ukrainian. Suppose we have to specify the verb ‘multiplizieren’ (for English is verb ‘multiply’). Following the name vp1 (verb, the present tense, subclass 1), the cortege of endings of the word class, to which this verb belongs, looks as follows:

```
Vp1: {-t,-t,-t,-en,-en,-en};
```

The corresponding names of the cortege positions (persons) are: er , sie, es, wir, sie, inf (in English, accordingly, he, she, it, we, they, infinitive). Note that some names of persons are skipped because they are typically not used in the professional texts.
So we are ready to specify the verb ‘multiplizieren’ referring it to mathematics:

```
      multiplizier => math : vp1 ;
```

In analytical languages the prepositions help to define the case of following nouns and to solve the problem of homonymy of endings. Next, we give examples of the specifications of German prepositions ‘mit’ and ‘neben’ (stands for, respectively, ‘with’ and ‘near’ in English):

```
mit => * : prep (dat) ;
neben => * : prep (dat, acc) ;
```

In parentheses, the case names (dative and accusative) are placed to indicate the cases of following nouns.

Given examples are intended to convey the idea of specification but not the troubles resulted from irregular nature of some words in every natural language. Particularly, if there is a need to perform an individual action related to irregular word, then the name of this action is placed into the specification of this word. The action itself must be developed in advance as a part of the LP.
From specifications DUAL generator builds dictionary and related data structures.

4. The LP generation process applications

The LP generation process for a natural language from its specifications, *sflex* and *slex,* includes the following steps:

Step 1. From specification *sflex* the LP MORF generates the computer presentation of endings *mflex* consisting of lists, tree-like structures, arrays etc. If the LP to be generated is intended for translation, then MORF builds two representations of endings of both languages using the corresponding specifications.

Step 2. The verification of specification *slex* and computer representation of endings *mflex* are performed with the help of LP GENW. It generates all forms of the words in the rule-by-rule order. The user analyzes the results and corrects them if necessary.
Step 3. From specifications *slex* and *mflex,* generator CON builds the monolingual or bilingual dictionary Ds of lexemes (stems with names of corresponding corteges) and the accompanying data structures. Then, the architecture of the LP generated looks as follows:

```
{ Ts, mflex, Ds }.
```

Here `Ts` is a configuration of components from *T* corresponding to the specification *slex*. LP CON facility can build new dictionary or add new stems to those that already exist. In particular, the same LP can alternate the use of several professional lexicons due to the possibility to join the specified words from different fields of knowledge with the common words that already exist in the dictionary. To include the professional lexicon of certain field of knowledge into dictionary means to specialize the LP in this field.

It should be noted here that the LP may process the text with empty dictionary of lexemes (but with *mflex*), when it is necessary to select the words from texts to include them to dictionary. In this case, the LP forms the frequency list of words to be specified.

With the help of DUAL generator two flexible LPs are built: DUET, for professional text translation from Russian into Ukrainian and from Ukrainian into Russian [Mi99], and FEST, for the statistical processing of texts in inflected and analytical languages [MS03]. DUAL generator and the LPs generated form the family of language processors cooperating around the dictionaries evolution in the course of text processing by any LP.

The LP DUET is restricted to word-for-word translation of highly specialized papers and reports. As a rule, translation with DUET of some professional text begins with generating frequency list of stems of words not found in dictionary. Then the specifications of words selected from frequency list are developed and dictionary is supplemented with terms specified. In such a way the dictionary becomes special in certain field of knowledge. Usually it contains no more than 10000 stems of significant words resulted in high speed of searching for the words during translation. It is successful due to the inflexibility and, what is more important, due to the similarity of syntactic phrase structure of texts in inflected languages, such as Russian and Ukrainian.

In the inflected languages, it is easy to recognize the agreement among adjacent input words (phrases) using the result of the morphological analysis. Due to the syntax similarity of the languages this agreement is transferred to the corresponding output words. This means that word-for-word translation is context-sensitive.

For inflected languages with similar syntax, such as Ukrainian and Russian, the result of translating professional texts shows that more then 85% of the output (for texts of good style up to 95% of the output) does not need human editing. In fact, this percentage shows that significant part of translation can be specified formally and processed by universal components of DUET. The rest should be translated by special software components. We decided not to complicate DUET with many special components. In this case post-editing performed manually by the user is necessary.

The comparison of output texts produced by LP DUET and existing commercial LPs of similar range shows at least two advantages of the LP DUAL related to the dictionaries:

- creative forming of lexeme specification assists in more correct choice of output lexemes;
- selecting terminology from the user's texts excludes the homonyms of lexemes from other field of knowledge.

The LP FEST performs morphological analysis of scientific and technical texts and forms the frequency lists of stems, or lexemes, or phrases, depending on user's request. Such frequency lists are used in many applications, in particular, as a first step in elicitation of subject matters of professional texts. Our main intention was to answer the questions how to use the statistic data to determine the subject of the text, for example, in order to classify the texts accordingly to their subjects, and to determine the relevance of the text to the subject defined by the terms presented in vocabulary. Usage of frequency lists to achieve these goals is based on the high frequency of terms in analyzed texts (usually from 20% to 35% of entries of terms). Only auxiliary words (conjunctions, prepositions, etc.) have higher frequency in scientific texts.

There are many ways how to recognize the terms in the texts. Some of the ideas are considered in [Du02]. Statistical analysis is commonly adopted as the base for terms elicitation by the user or in cooperation with him/her. With the frequency list of lexemes or phrases, the user needs to analyze only the upper part of the frequency list, which is usually rather small (from 10 up to 30 lines), instead of the whole text.

The LP FEST forms the frequency lists in the following steps:

Step 1. The text is analyzed by FEST using only endings and the dictionary of auxiliary words. The task is to generate the sequence of words not found in the dictionary. Each word should be presented in the sequence by its stem and ending (if any). Note that because of absence of stems dictionary the false endings may be identified in some words.

Step 2. The frequency list is built for the stems that were not found in the dictionary. Each stem is associated with the endings of all word forms, which begin with this stem.
The user analyzes the frequency list of stems. He may correct the stems with false endings.
As a result, two alternative conclusions can be made:

a) frequency list of stems can be used to recognize the subject matter of the text, so the goal is achieved;
b) frequency list of stems doesn't contain enough information to understand the subject matter of the text. In this case, to make the information about text more convincing, the user can make a request to build the frequency list of phrases containing words selected from upper part of frequency list. To do this, the dictionary of such words must be generated.

Step 3. The user forms monolingual specification of the words selected (the LP for generating the specifications using the frequency list of stems is currently under development).

Step 4. The LP GENW generates the word forms using specifications formed in the previous step. The user analyses word forms generated and corrects wrong specifications.

Step 5. The text is analyzed by FEST using endings and dictionary of auxiliary words (optionally) and newly compiled dictionary of meaningful words. After this, the frequency list of phrases is formed. The user analyses the frequency list of phrases and makes a final decision about the subject of the text.

To determine if the text belongs to the certain field of knowledge, the user has to analyze the frequency list of lexemes or/and frequency list of phrases of words formed by FEST using the dictionary of terms of this field of knowledge.

5. Conclusion

In this paper, the methods of formal specification are applied to the processing of professional texts in the natural inflected languages. For this application, we developed the methodology and the architecture for the languages processor generation. User-oriented formal language for the specifications of the morphology and translation of inflected languages are developed together with the tools for dictionary generation from formal specifications. Two language processors are generated: DUET, for word-for-word translation of professional texts in inflected languages, and FEST, for statistical processing of texts in inflected and analytical languages using Latin and Cyrillic alphabets.

Experiments with the resulting language processors show the usefulness of the implementation of the formal tools for specification of word-by-word processing of texts in natural languages. With the generator DUAL, user performs only textual specifications of vocabulary of his/her field of knowledge. The advantages of the LPs considered are flexibility and convenience for the initial processing of professional texts.

The LP DUET is used locally to translate texts related to specific areas of computer science. The improvement is achieved by employing the user-oriented vocabularies.

The LP FEST can process inflected and analytical languages that use Latin and Cyrillic alphabets. It can be improved at the expense of new software components responsible for further refinement of phrase selection using the dictionaries of words and abstract notions, which identify the classes of words and the relations among them.

Bibliography

[Ba97] Batory, D.: Intelligent Components and Software Generators, TR-97-06, University of Texas at Austin, 1997.

[Do02] Doroshenko, A.; Kuivashev, D.: Making intelligent retargetable optimizing compilation for DSP microprocessors. "Problems in Programming", N 1-2, 2002; pp. 477-488.

[Du02] Duda, O.: Semantic analysis of corpus of special texts. Problems of Ukrainian terminology. Proc. of Lviv National University “Lvivska Politekhnika”, 2002, 453; pp. 296-300.

[Go02] Collberg, C.S.: Automatic Derivation of Compiler Machine Descriptions. ACM Transaction on Programming Languages and Systems, 2002, 24(4); pp. 369–408.

[Kr92] Krueger, C.W.: Software reuse. ACM Computing Surveys, 1992, 24(2); pp. 131-183.

[Mi00] Mishchenko, N.M.: On language processor generation from formal specification of domain-specific vocabulary. In Proc.Int. Workshop on Computational Linguistics and Its Applications Dialog'2000, Protvino, Russia, June 2000. Vol. 2 of 2. Applications. Protvino 2000; pp. 271-278 (in Russian).

[Mi98] Mishchenko, N.M.: About reusability of language processor generator tools. In Proc. of UkrPROG: Int. Scientific and Practical Conference on Programming (Sept. 1998, Kyiv), 1998; pp. 462-467 (in Ukrainian).

[Mi99] Mishchenko, N.M.: A mobile word-for-word formal-specification-based translation system for scientific texts in inflected languages. Cybernetics and System Analysis, 1999, N 1; pp.33-42 (in Russian).

[MS03] Mishchenko, N.M.; Shchogoleva, N.M.: On lexical and statistical analysis of scientific and technical texts. In Int. Conf. "Knowledge-Dialogue-Solution" (16-26, June, 2003, Bulgaria) (in Russian) – accepted for publication.