# AUTOMATED IMPROVING AND FORMING WORDNET SYNSETS ON CONVENTIONAL (NON COMPUTER BASED) SYNONYM AND BILINGUAL DICTIONARIES

G. Totkov, P. Ivanova, Iv. Riskov

## Abstract

WordNet is an on-line lexical database of English language. Word meanings are represented by synonym sets (synsets). Each synset corresponds to one lexical concept. Different lexical and semantic relations link the synsets. An algorithm for automated forming Bulgarian synsets, corresponding to WordNet synsets, is presented. The three main steps of the algorithm are: I Automated improving of synsets on synonym dictionary that includes discovering synsets representing one concept; synsets which contain words for two or more different concepts; synsets with missed or incorrect synonyms placed in them, etc.; II. Finding a correspondence between lines of Englilish-Bulgarian Dictionary and WordNet synsets; III. Forming Bulgarian synsets coresponding to 55,000 WordNet concepts, using results from steps I. and II.

## Introduction

The building block of WordNet (WN) [2, 3] is a synonym set (synset) of words that expresses a given concept. WN gives definitions (explanatory glosses) and sample sentences for its synsets. The basic semantic WN relations are hyponymy (X is a kind of Y), hypernymy (this is a kind of X), meronymy (part of this X), holonymy (this is a part of X), entailment for verbs (like meronymy for the nouns), antonymy, and synonymy. The project to create EuroWordNet as a multilingual database with wordnets for several European languages was completed in 1999 [9] and continued in 2002 [7]. In addition, the wordnets are linked to an Inter-Lingual-Index, so that it is possible to go from the words in one language to similar words in any other language.

The main purpose of the paper is to describe a method and tools for semi-automatically mapping of Bulgarian synsets to English WordNet (EWN) entries and, therefore, building a Bulgarian WordNet (BWN). The first attempt to build a core of BWN (only for nouns) using semi-automatic extraction from the original EWN is described in [5].

In the process of building BWN we used the following resources:

- original EWN dictionary – over 175,000 entries – contains information of English words (actually one of the meanings of that word), which consists of the word itself, its synonyms, glosses and WN identifier number;
- English-Bulgarian Dictionary (EBD) – about 165,000 entries;
- Bulgarian Synonym Dictionary (BSD) [4] with more than 37,000 synonym paradigms.

Our first goal was to convert all three dictionaries into tab-separated plain text format, where each line corresponds to one entry in the dictionary. All fields in the dictionary that are not needed are stripped, so that less effort is needed to process the dictionary in the next stage.

The process of automated improving and forming BWN on conventional BSD and EBD may be split into **three steps** that are fairly independent.

## I. Automatic improvement of standard synonym dictionary

The BSD contains synonym rows (SR) each of which includes the synonyms for one meaning of a given lexeme. The lexeme itself stands in the left-hand part of the row and is called leading lexeme (LL) while its corresponding synonyms

occupy the right-hand part of the SR. A combination of SR with one and the same LL will be called synonym paradigm (SP) of the lexeme.

Example 1. Several synonym rows from BSD

администрация - управление, управа, ръководство

администрация - чиновничество

администрирам - управлявам, ръководя, завеждам2

администрирам неодобр. - вж. началствам

администрирам - държа кормилото книж.

активизирам - вдигам (турям, поставям) на крак (нога) разг.

The first step includes discovering different synsets representing one concept; synsets, which contain words for two or more different concept (mixed synsets); synsets with missed or incorrect synonyms placed in them, etc.

In the right-hand of SR, the lexemes can be whole phrases and/or a reference to another LL by means of *see* (вж.) can be introduced as well. A lexeme can be associated with characteristics that describe the scope or emotional colouring of its usage or imply certain grammatical features.

In view to locate and remove various types of errors, gaps and discrepancy that have been made during the dictionary composition, we undergo several stages of SRs processing [8].

### *Removal of spelling mistakes and localization of omitted synonym rows or lexemes*

1. Spell-checking the correct abbreviation denoting the usage scope and the emotional colouring of words.
2. Expanding the contracted records given in brackets (see the last line of Example 1) when recording synonym phrases or when a second form of the lexeme is given. To precise whether both forms are used in the given meaning estimation is to be made if in some rows the two forms are both given while in others only one of the forms is given.
3. Localizing lexemes that can be found only in the right-hand (or left-hand) part of the SR. The reasons for this may be as follows: wrong spelling of the lexeme in some of the SR (either as LL or when found in the right-hand part of the SR); omitted synonym rows for some of the lexeme meanings. In the latter case, a suggestion for adding a row must be made. It can be checked whether a combination of a lexeme and its characteristics is not found only in the left-hand or in the right-hand part either.
4. Expanding reference rows where in the synonym paradigm of the reference-lexeme such a row is sought that contains the LL of the row in the right-hand part of the reference-row. The following cases may occur: no such a row – the reasons are similar to those in 3); only one row – in the right-hand part of the reference-row all lexemes from that row except the one which is leading in the reference-row are recorded; more than one row – for each of these rows an equivalent row must be added in SP of the reference-row (if such a row does not exist), i.e. the reference row is expanding in several rows. It is possible to employ hand selection of row(s) for expanding.

### *Removal of logical discrepancies*

The synonym row contains synonyms of one meaning of a given lexeme, i.e. SR defines one concept. The dictionary of synonyms lack contradictions if each synonym row made of $n$ lexemes, contains another $n-1$ synonym rows, made of the same lexemes. The number of SRs containing the same $n$ lexemes is exactly $n$, provided none SP is allowed to contain similar rows. Therefore, the dictionary of synonyms lacks contradictions if a concept is denoted by means of $n$ lexemes, the dictionary must contain $n$ SR, in which the leading lexemes are each of those $n$ lexemes and that are made of the same $n$ lexemes.

At this stage a verification whether the SP of each word in the right-hand part of any SR contains a row equivalent is to be found. The objective is to edit the rows for the different meanings of the lexemes in a way that the row description should be complete and should lack contradictions. We shall consider that a row description is contradictory if the SP of at least one lexeme in its right-hand part does not contain an equivalent row.

All rows whose description is not contradictory we may mark as processed and those rows are not modified during the further processing.

After the first stage in the processed BSD there are 37,008 SP; rows with phrases – 1,666; logically non contradictive – 14,609; automatically extracted concepts – 4,344; unprocessed– 22,399.

For the next unprocessed row, we should find the relevant one (that refers to the same concept) in the SP of the word in its right-hand part. For example, a bit wider combination M of all rows which have at least one common lexeme with the given. This combination contains all rows of the SP of the lexemes of the row discussed. Due to the operations which are

performed later we must not exclude those SP that have been already processed. Since in the row there may be no lexeme, we include in M also those rows where the leading lexeme is not found in the discussed row but has common lexemes with it.

The less a row differs from the one that is being processed at the moment, the greater the chance that it refers to the same concept. That is why we estimate the rows of M by their distance to the discussed row.

Let denote by $d(R_1, R_2) = \frac{|R_1 \setminus R_2| + |R_2 \setminus R_1|}{|R_1 \cup R_2|}$ the distance between the rows (sets) $R_1$ and $R_2$ and by $SP_R^k$ – the SP of the k-th lexeme of the synonym row R. Let denote by $d(R, SP_R^k)$ the shortest distance between a row $R$ to a row of a SP of its k-th lexeme, i.e. $d(R, SP_R^k) = \min\{d(R, R_i) : R_i \in SP_R^k, i = 1, 2, ..., |SP_R^k|\}$. The evaluation of $R$ shall be called the number $d(R) = \frac{1}{n}\sum_{k=1}^{|R|} d(R, SP_R^k)$.

In BSD the following inaccuracies may occur when compiling and editing of synonym rows and they may be the reasons for the contradictive descriptions of a concept or the lack of an equivalent row in some SP of the words in the right-hand part of a row: a) denoting a concept by means of more than one row– such rows must be merged into one; b) denoting more than one concept by means of one row ("merging of rows") – such a row must be divided into two (or more) rows; c) omitted, added, or misspelled lexeme within the row – such a word must be added, erased or corrected (the misused one should be erased and the right one should be added); d) omitted row– such a row must be added.

In the experimental program for splitting/merging synonym rows (S/M_SP program), for each row R), whose evaluation is greater than 0, the rows comprising the combination M are displayed sorted by their distance to R (see Fig. 1).

When the expert examines them he chooses one together with R and is to apply one of the following activities: a) *unification*—in this case each row is being completed with the lexemes that the other lacks; b) *division* – in this case each of the row is being divided into two rows with one and the same LL and containing the section and the difference of the combinations of their lexemes; c) *movement* of lexemes from one row to another; d) *deletion* of lexemes.

For those rows whose evaluation is fairly small, a unification can be made of this row with the nearest rows of the SP of each word on the right and this unification can be offered for approval as SP describing a concept (eventually an exclusion of lexemes may occur). The unification row can also be evaluated and the approval of unification can be automatically requested in case of fairly small evaluation.

An experiment has been carried out for automatic handling of SP form the dictionary in case of evaluation less than 0.2 and evaluation of the unification less or equal than 0.5. Using this method contractions in 14,717 rows have been fixed referring to the description of 2,252 concepts. The remaining unprocessed rows are estimated to one sixth of those which do not contain phrases.

### *Numbering of the lexeme concepts in rows not containing phrases*

All occurrences of a certain lexeme in the dictionary as a leading one are found consecutively by means of an integer from 1 to the number of its occurrences. Then the lexemes in the right-hand part of the row take the same numbers as they have as leading in the equivalent row of their SP. If at Stage II. the processing has ended successfully (the dictionary is non-contradictive), then such and equivalent rows must be found in each SP of a lexeme of the given row. The same lexemes in the equivalent rows must have the same numbers.

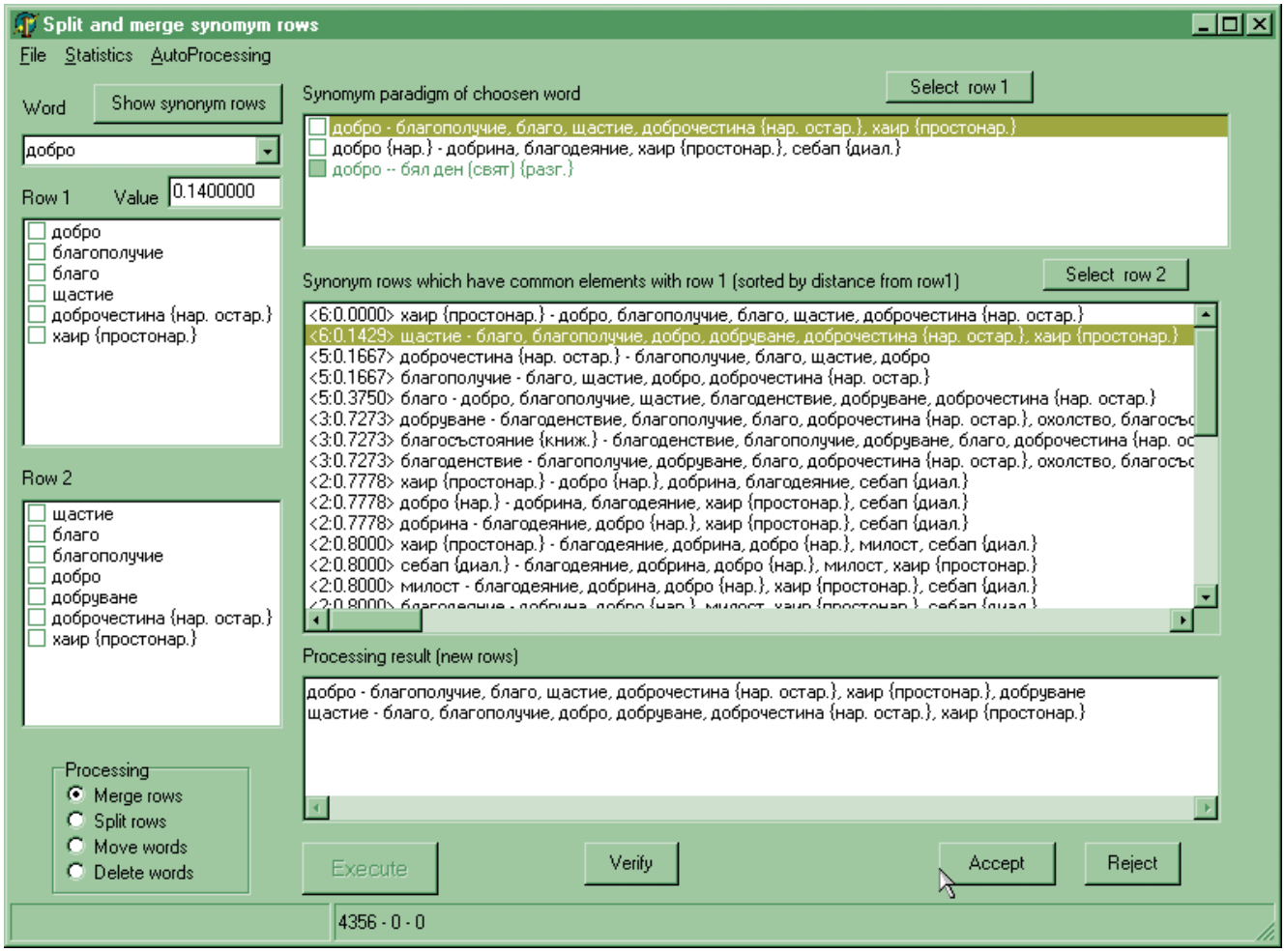



Fig. 1. Main menu of the S/M_SP program

*Association of the phrase rows with the rows corresponding to the same concept.*

For some LL, the SDS contains separate SP only of phrases in the right-hand part. Those phrase rows usually supplement another SP of their leading lexeme.

The latter make is difficult for the automatic estimation of the lexeme meaning that they are related with.

It is possible that some SP with phrases in the right-hand part to refer to meanings of LL expressed with another SP but it is also possible, although in rare cases, to refer to a meaning for which there is no a SP consisting only of words (the given LL in this meaning has no synonyms). It is necessary in this case to find each phrase (or the combination of phrases occurring in the right-hand part) of the SR with which LL take part and whether those lexemes form a synonym row for some of their meaning.

## II. Finding a correspondence between rows of EBD and WordNet synsets

The assignment is to find the corresponding Bulgarian exact translation in the EBD for every entry of the English WN. The task is practically to match each WN row with its respective EBD one. With this view, each WN line is provided with an identification number, which is transferred into a special field in the EBD.

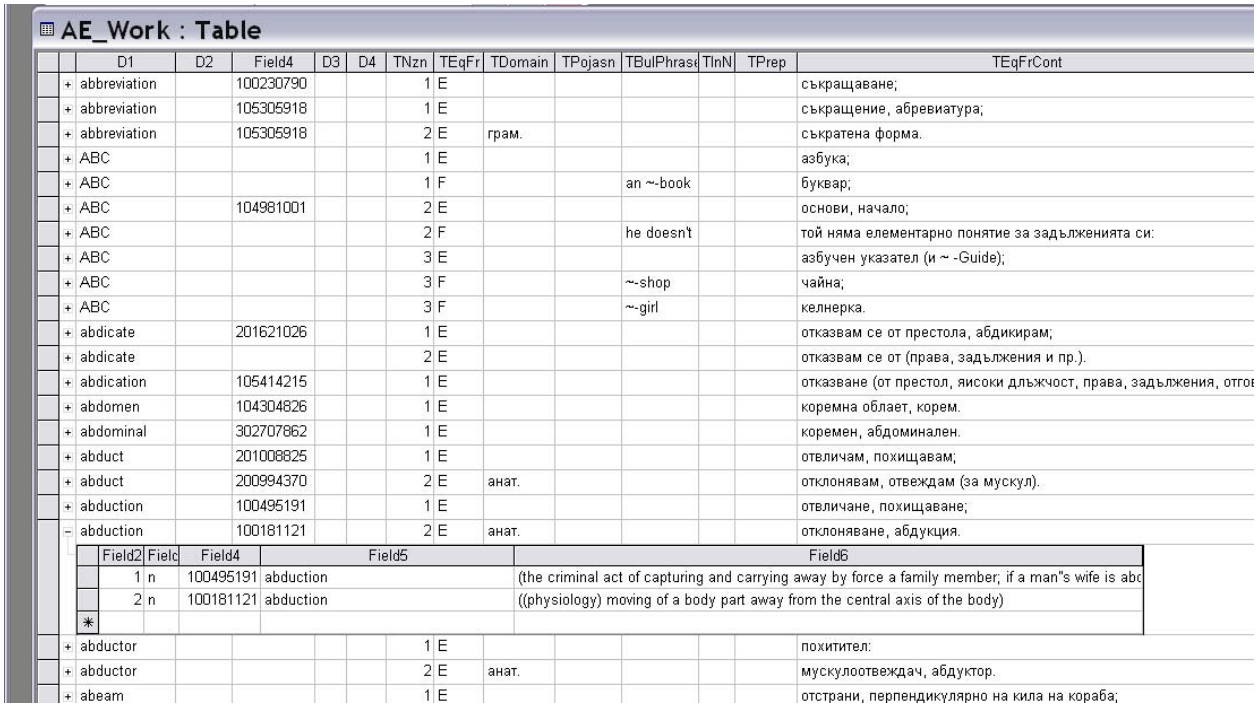
The problem is tackled with by the creation of two Access tables – the first one represents EBD and the second one – EWN. Then the EBD rows are joined with the EWN rows so that the joined fields from both tables are the corresponding English words.

The so-related tables are processed by professional translators who transfer WN numbers to the corresponding column in EBD table (Fig. 2). In case that a Bulgarian signification is missing, it is then added as a new row in EBD with a respective number and translation. Sometimes a particular Bulgarian synonym nest has to be divided into two separate rows in order to precise the nuances in the English word or to sever reflective from non-reflective verbs (given in brackets in the Bulgarian case).

For example, the need to create new rows in EBD arises when an animal or botanical issue appears because Bulgarian language construes a set of words for a single English one (for instance partridge flesh and the partridge bird itself for the single partridge or the apple tree, apple wood and the fruit apple for the English apple). New inventions and notions from the fields of the computer science, technology, culture etc. require finding Bulgarian translations as well.

On the other hand, a substantial quantity of the Bulgarian number fields remains blank, without an English correspondence one. The cause might be older meanings or phrasal verbs, which tend to miss from the WordNet.

To sum up, the result is the formation of correspondence between the WN and Bulgarian entries. The additional effects are the enrichment and updating of the EBD and the synchronization of the lexicography methods. As a result more than 55,000 WN synsets are juxtaposed to the corresponding rows in EBD.



AE_Work : Table

| | D1 | D2 | Field4 | D3 | D4 | TNzn | TEqFr | TDomain | TPojasn | TBulPhrase | TInN | TPrep | TEqFrCont |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| + | abbreviation | | 100230790 | | | 1 | E | | | | | | съкращаване; |
| + | abbreviation | | 105305918 | | | 1 | E | | | | | | съкращение, абревиатура; |
| + | abbreviation | | 105305918 | | | 2 | E | грам. | | | | | съкратена форма. |
| + | ABC | | | | | 1 | E | | | | | | азбука; |
| + | ABC | | | | | 1 | F | | | an ~-book | | | буквар; |
| + | ABC | | 104981001 | | | 2 | E | | | | | | основи, начало; |
| + | ABC | | | | | 2 | F | | | he doesn't | | | той няма елементарно понятие за задълженията си: |
| + | ABC | | | | | 3 | E | | | | | | азбучен указател (и ~ -Guide); |
| + | ABC | | | | | 3 | F | | | ~-shop | | | чайна; |
| + | ABC | | | | | 3 | F | | | ~-girl | | | келнерка. |
| + | abdicate | | 201621026 | | | 1 | E | | | | | | отказвам се от престола, абдикирам; |
| + | abdicate | | | | | 2 | E | | | | | | отказвам се от (права, задължения и пр.). |
| + | abdication | | 105414215 | | | 1 | E | | | | | | отказване (от престол, яисоки длъжчост, права, задължения, отгов |
| + | abdomen | | 104304826 | | | 1 | E | | | | | | коремна облает, корем. |
| + | abdominal | | 302707862 | | | 1 | E | | | | | | коремен, абдоминален. |
| + | abduct | | 201008825 | | | 1 | E | | | | | | отвличам, похищавам; |
| + | abduct | | 200994370 | | | 2 | E | анат. | | | | | отклонявам, отвеждам (за мускул). |
| + | abduction | | 100495191 | | | 1 | E | | | | | | отвличане, похищаване; |
| − | abduction | | 100181121 | | | 2 | E | анат. | | | | | отклоняване, абдукция. |

| | Field2 | Fielc | Field4 | Field5 | Field6 |
|---|---|---|---|---|---|
| | 1 | n | 100495191 | abduction | (the criminal act of capturing and carrying away by force a family member; if a man"s wife is abc |
| | 2 | n | 100181121 | abduction | ((physiology) moving of a body part away from the central axis of the body) |
| * | | | | | |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| + | abductor | | | | | 1 | E | | | | | | похитител; |
| + | abductor | | | | | 2 | E | анат. | | | | | мускулоотвеждач, абдуктор. |
| + | abeam | | | | | 1 | E | | | | | | отстрани, перпендикулярно на кила на кораба; |

Fig. 2. EBD and WND as two joined Access tables

# III. Forming Bulgarian WordNet

The **third step** in the process of setting up the corresponding BWN requires Bulgarian synonyms to be separated in each row of EBD in the translated Bulgarian equivalent of the given English word. Any such line contains a list of meanings for the target word, separated by commas. However, some of the meanings may not be a single word, but a phrase and that phrase could in turn contain commas, which are not to be treated as separators. Our goal is to design an efficient algorithm that determines which of the commas are separators for the different meanings and which are not such kind of separators. To support our thesis we provide an implementation of the algorithm. The one we are discussing here is designed mainly for EBD, although its ideas could be used successfully for other dictionaries. The structure of the EBD entries varies, as well, but at least it contains the description of the word that the entry refers to. Such descriptions are usually a list of words or phrases delimited by commas and terminated by semicolons. The problem that arises is that when there is a phrase, it is not clear if a comma separates this phrase from the next one or is it just part of the phrase. This is useful when we are trying to separate the different meanings in the description.

Here are some examples taken from the dictionary we use. All of them contain commas, which are not used as meaning delimiters: a) "човек, който може да действува съобразно със собствената си воля" (a man who can act in accordance with his own will); b) “поддържам по-благоприятното становище, вярвам, че доброто ще победи” (I support the more favourable opinion, I believe that the good will prevail); c) “болничен служител, натоварен със социални грижи за болните след изписването им” (a medical officer, charged with the social care for the patients after their discharge from hospital).

Taking a closer look at the dictionary, we may notice that there are descriptions that contain no special symbols (delimiters like commas, brackets, etc.) at all, so we could treat them as a single word or phrase. This may reduce the number of unprocessed entries. We may even skip entries that contain special characters except commas, since we consider only

comma processing here. Having made both of the above operations on our dictionary, we reduced the total number of entries by 35%.

After that initial step we are now ready to get to the real processing. We do this in three steps for each entry:

1) We find all positions of the commas in the description.

2) For each comma we determine if it is a meaning delimiter or not. We do that by applying a set of rules. These rules try to look at the words, surrounding the currently processed comma. They are based on common language patterns. Each rule returns one of the following results: the rule cannot determine the role of the comma; the rule determines the role of the comma as a meaning delimiter; the rule determines the role of the comma not as a meaning delimiter. Rules are processed until one of them could determine the role of the comma (actually, the last two results). The order of the rules does matter, since it is possible that a general rule will override a more specific one. By applying the correct order (i.e. all general rules are processed last) this would not happen.

3) It is possible that no rule could determine a role of a comma after that all rules have been applied. In such case, we keep a list of exceptions. Each time the above happens we put the description in the list. After all entries are processed this list is revised and a new rule, that matches as many of the exceptions, as possible, is made. It is then added to the list of the rules and the whole process is repeated once again.

We do this until the exception list is either empty, or it contains enough entries to be processed manually.

It seems that the rules are a key part in our algorithm. A properly written rule is the one that matches correctly as many entries, as possible. Thus, it is necessary to provide enough information to the rule. We decided to do this by adding grammatical characteristics to each word in the description. The information we included consists in the grammatical class the word belongs to, the base form of the word, the tense of the word if it is a verb, and many other characteristics that might be helpful.

The implementation of the algorithm could be split up into several parts: defining the data structures; the routines that determine if the whole description needs further processing i.e. does it contain a comma; a comma splitting routines; a rule interpreter; exception list maintaining routines.

The data structures and the rule processing are the essential parts. Because the entries are processed one at a time there is no need for a data structure to hold all the entries. What is important here is the grammatical information about the words in the description. For each word, we need to store its base form, its grammatical class and its properties. In fact, when we have this information we do not need the real word. The relation "word – grammatical information" is uniquely defined. A good data structure would keep the base form of the word in form of a string, the grammatical class as an identifier and the properties as flags. Apart from this, we need information for each comma whether it has been determined as a separator or not.

There are two ways for the rules to be stored. They could be hard-coded into the processing program or they could be implemented as scripts run by a special interpreter. We decided to implement the second method. So we designed a simple language, based on the postfix notation. This language, as every other postfix based language, consists of operands and operators. Operands could be strings, numbers or special constants that refer to the current task e.g. one such constant is the position of the currently processed comma. Operands are stored in stack, which is the only data structure our interpreter uses. Operators, in turn, are small routines that are hard-coded in the interpreter. They are fixed and, unlike rules, do not need to be changed. These operators take their operands from the stack and return their result back in the stack. To make a better understanding of how a rule looks, we provide some examples of rules we use:

1) A simple rule that matches a word that usually appears after a comma and is not a meaning delimiter:

```
comma_pos word[]
“че” =
if
  2 return
endif
1 return
```

2) A simple rule that matches many words and each of them is usually preceded by comma, when used

```
comma_pos word[]
“който” =
```

comma_pos word[]

“която” =

comma_pos word[]

“което” =

comma_pos word[]

“които” =

or or or

if

2 return

endif

1 return

# Conclusions

The automated linking of the BSD with the already created BWN is forthcoming. For this end from the two lexical resources one synonym dictionary can be formed, which the software tools for ameliorating the monolingual synonym dictionaries created in Step I can be applied on. The resulting synonym dictionary not only would improve the BWN but moreover would complement the original EBD because of its worked out links with the BWN.

The construction of BWN, comparable in scope with the EWN, would require the supply of additional lexical resources and the creation of new software tools. The digitalization of the Explanatory Dictionary of the Bulgarian language [1] and the Bulgarian-English Dictionary is about to be finished and procedures for their automated alignment with the improved EWN are being worked out. As a result, the Bulgarian glosses will be automatically attributed to their respective synonym lines in the Bulgarian corpus and a completely mirrored Bulgarian-English-Bulgarian dictionary will be created. In plus, new Bulgarian words will be added to the already existing synsets in BWN.

A problem which remains open to this stage is the usage of derivative links between separate words (synsets respectively) in BWN and EWN. From the word formation point of view various suffixes/affixes tend to alter the word radix in Bulgarian language but these transformations are not presented in the electronic dictionaries we used. So, in Bulgarian synsets, we miss a lot of adverbial nouns because of incomplete dictionary input.

Presently, we are carrying out a preliminary preparation for the utilization of Word Formation Dictionary of Contemporary Bulgarian [6] for enlarging BWN automatically.

# References

1. Adrejchin L. et al. Bulgarian Explanatory Dictionary (in Bulgarian)// Sofia. Nauka I Izkustvo, 1999.
2. Fellbaum C. (ed.). WordNet: An Electronic Lexical Database//The MIT Press, Cambridge, London, England, 1998.
3. Miller G. A. et al. Introduction to WordNet: an on-line lexical database//International Journal of Lexicography 3 (4), 1990.
4. Nanov L, Nanova A. Bulgarian Synonym Dictionary (in Bulgarian)//Sofia. Nauka I Izkustvo, 1987.
5. Nikolov T. Building a Core for Nouns for Bulgarian WordNet (in Bulgarian)//Diploma thesis, Sofia University, 2000.
6. Penchev J. (ed.) Word Formation Dictionary of Contemporary Bulgarian (in Bulgarian)/ Под. ред. Й.Пенчев. София, 2000.
7. Stamou S. et al. BALKANET: A Multilingual Semantic Network for the Balkan Languages//Proceedings of the International Wordnet Conference, Mysore, India, 21-25 January 2002, 12-14.
8. Totkov G., Ivanova P. Automated Improving and Forming Synsets on Conventional (non computer based) Synonym Dictionaries//International Conf. Automation and informatics’2002, 5-6 Nov. Sofia, 33-36.

9. Vossen P. Building a multilingual database with wordnets for several European languages//1999, http://www.hum.uva.nl/~ewn.