SEMIAUTOMATIC GENERATION OF WORDNET TYPE SYNSETS AND CLUSTERS USING CLASS METHODS. AN OVERVIEW

FLORENTINA HRISTEA

Abstract. As its authors note, Miller *et al.* (1990), WordNet (WN) is a lexical knowledge base, first developed for English and then adopted for several Western European languages, which was created as a machine-readable dictionary based on psycholinguistic principles. The present study is an attempt to discuss the semiautomatic generation of WNs for languages other than English, a topic of great interest since the existence of such WNs will create the appropriate infrastructure for advanced Information Technology systems. Extending the algorithmic approach introduced in Nikolov, Petrova (2001), we propose a semiautomatic method based on heuristics for the generation of WN type synsets and clusters. The focus is on noun and adjective synsets, since nouns and adjectives have completely different organizations in WN, but verb and adverb synset generation is also addressed. The target language for performing tests will be Romanian. Our approach to WN generation relies on so-called "class methods", namely it uses as knowledge sources individual entries coming from bilingual dictionaries and WN synsets, but at the same time demonstrates the need to combine such methods with structural ones.

1. INTRODUCTION

WN has been recognized as a valuable resource in the human language technology and knowledge processing communities. The human language research community has encouraged the development of WNs for languages other than English, at the same time concentrating on the possibility of automatically generating such huge lexical data bases. The main reason for this is the desire and the necessity to create **a uniform ontological infrastructure across languages**. This can be achieved since, while concepts are language dependent, the basic set of relations that link the concepts remains the same. This means that the inference algorithms for extracting information remain the same. The existence of such an uniform ontological infrastructure across languages will therefore simplify machine translation from a language to another and will facilitate the use of the same reasoning schemes and algorithms developed in conjunction with the American WN.

RRL, **LII**, *1–2*, p. 97–133, Bucureşti, 2007

The present study concentrates on the important and up-to-date topic of automatic generation of WNs for languages other than English. The approach to WN generation consists of a semiautomatic method based on heuristics which belongs to the so-called "class methods" Atserias *et al.* (1997). It therefore uses individual entries coming from bilingual dictionaries and WN synsets as knowledge sources, and hence the success of our method depends directly on the availability of comprehensive bilingual dictionaries in electronic format.

The basic translation algorithm (Algorithm 2.1 of the present paper) will be using the so-called "elementary sets", a concept introduced in Nikolov, Petrova (2000). Algorithm 2.1, which is described in Nikolov, Petrova (2001), will be further completed by Algorithm 2.2, proposed in Hristea (2002), which performs a backtracking action (step 1) in order to obtain as final output the foreign synset corresponding to the given English one. It should be noted that the Bulgarian authors who first describe Algorithm 2.1 Nikolov, Petrova (2001), having as output a sorted list of elementary sets, make no comment whatsoever as to how they obtain the final foreign synset, in their case the final Bulgarian noun synset. One can easily assume that it is manually obtained by linguists using the output of Algorithm 2.1. It was the concern of Hristea (2002) to automate the process of creation of a foreign WN type synset to the largest extent that this is possible, and our comments concerning output obtained in the case of the Romanian language will be made within this type of framework[1].

Since the same Bulgarian authors Nikolov, Petrova (2001) do not specify what evaluation function has been used, additional comments will be made here, taking into account the mentioned Romanian output, with respect to the type of evaluation function that was or should be used in the translation process.

Finally, to the praise of the mentioned authors, who are only concerned with obtaining "a core of Bulgarian noun synsets", it turns out that their algorithm can be extended (more or less successfully) to the general case of any foreign language (not just Bulgarian). Additionally, it is our belief that Algorithm 2.1 can be successfully used in the case of all other (three) parts of speech that WN deals with, provided that it is modified accordingly. Such modifications should take into account the typical semantic relations implemented in WN with regard to each part of speech, thus combining the class method initially used in the case of nouns with a structural approach to WN generation (see the various enrichment techniques proposed in the present study).

Since in WN adjectives have a completely different organization than nouns – the N-dimensional hyperspace – our study concerning this part of speech is further extended by taking into account the semiautomatic generation of foreign adjective clusters. At this point our approach again makes the necessary links between class methods and structural ones (namely those that take profit of the WN

[1] And using **version 2.0** of the WN database.

structure). Algorithm 3.1, proposed in Hristea (2002), represents a first approach to semiautomatic generation of foreign adjective clusters which does not make use of monolingual resources but only of bilingual ones, namely bilingual dictionaries in electronic format.

2. THE TRANSLATION ALGORITHM

The algorithm for translating a given English synset into the corresponding synset in a language other than English will be using so-called "elementary sets" or **e-sets**, a concept introduced in Nikolov, Petrova (2000). An e-set corresponds to a monosemous reading (sense) of a word and can be defined as follows:

Definition 2.1

An *e-set* relative to a word is the set of synonyms corresponding to a specific monosemous reading (sense) of that word.

Let us denote by EW any English word and by FW any foreign word, namely a word of a language other than English. Let **eword** of sequence (1) be an EW, while *fword1, fword2* and *fword3* of the same sequence are its corresponding translation equivalents (according to the appropriate bilingual dictionary):

eword *fword1; fword2, fword3* (1)

In order to distinguish among *fword1*, *fword2* and *fword3* two different separators are used in standard paper dictionaries. A semicolon separates different meanings of a given word. A comma separates synonyms which refer to one and the same meaning of the word. (In this case *fword2* and *fword3* are synonyms). This is the form of a bilingual dictionary which will be used by the programs implementing the proposed translation algorithm. In the above example the involved e-sets are

$$\{fword1\} \text{ and } \{fword2, fword3\}.$$

The computer programs which implement the translation algorithm will generate the list of all e-sets of FWs corresponding to the meaning of all EWs occurring in a given English synset. The foreign synset corresponding to the studied English one is formed of one or more of the generated e-sets (which can be adjoined). The "candidates" for inclusion in the foreign synset are *labeled e-sets*, namely those e-sets which contain *labeled words*.

In order to label the FWs belonging to the generated e-sets, we have decided to first label the EWs belonging to the English synset. These EWs will be labeled with integer numbers ranging from 1 to n (where n is the size of the synset, namely the number of words it contains), in the order of their occurrence. After labeling the EWs of the original synset, the FWs of the generated e-sets are looked up in the corresponding bilingual dictionary. Each time an EW of the given synset represents

the translation, according to the dictionary, of a FW, the corresponding FW receives the label of that EW. If any word of a foreign e-set can be translated into a word of the English synset using the bilingual dictionary, the whole foreign e-set is moved to the "list of candidates". As noted in Nikolov, Petrova (2001), when completed, this list of candidates is the most important preliminary result. The appropriate foreign synset must be a compilation of some e-sets belonging to this list. Various *evaluating functions* which sort the extracted e-sets and outline the most adequate ones have been developed. In order to define such evaluating functions let us refer to the following concepts:

Definition 2.2

The *label of an e-set* represents the number of labels assigned to the words belonging to that e-set.

Definition 2.3

An e-set is *unlabeled* if it contains no labeled words.

Any word can have one or more labels assigned to it (as well as no label at all). The most common evaluating function which is proposed in the literature Nikolov, Petrova (2001) takes as argument an e-set and has a value given by the very label of that e-set. A variant of this evaluating function is that which divides the number representing the label of the e-set to the size of the same e-set.

As far as we are concerned, we have taken into consideration the evaluation function which is defined below.

Each EW belonging to the given English synset will have a label (represented by an integer number from 1 to n, where n is the size of the synset) and the labeling of the FWs belonging to the e-sets is performed according to this label. The labels of the foreign words which differ from the label of the corresponding EW will be considered as representing two points, while the others represent just one point. The value of the evaluation function relative to a specific e-set is given by the total number of points corresponding to that e-set divided by its size.

Having defined all necessary concepts, one can now state the algorithm for generating the foreign *e-sets* corresponding to a given English synset:

Algorithm 2.1

Input: The file containing the English synsets and the two files representing the two bilingual dictionaries (for instance, the English-Romanian and the Romanian-English dictionary respectively).

1. Create (by consulting the appropriate bilingual dictionary) the e-sets corresponding to each word of the given English synset.
2. Label the English words belonging to the given English synset.
3. Label each of the e-sets generated in Step 1.

4. Remove all unlabeled e-sets.
5. Evaluate the e-sets (using the assigned labels and an evaluating function).

Output: The sorted list of e-sets corresponding to the given English synset.

The translations in the foreign language of the words occurring in the English synset are extracted from the bilingual dictionary as follows:

eword*1* meaning*11*; meaning*12*; … ; meaning$1m_1$
……..……
eword*n* meaning*n1*; meaning*n2*; … ; meaningnm_n

The set of e-sets generated by Algorithm 2.1 is of the following form:

$$\{\{\text{meaning}ij\} \mid 1 \leq i \leq n, 1 \leq j \leq m_i\}.$$

The *foreign synset* will be generated using this set.

In the automatic generation of the *foreign synset* corresponding to a given English synset we shall also take into account

Remark 2.1

Of all possible meanings of a word, only one refers to a specific concept (to which a synset corresponds).

Using the sorted list of e-sets generated by Algorithm 2.1 (namely the evaluated e-sets), the meaning (elementary set) evaluated with the highest value will be chosen corresponding to each English word. Let this meaning, corresponding to *ewordj*, be $meaningji_j$. The *foreign synset* will be generated using the e-sets obtained by means of Algorithm 2.1, taking into account Remark 2.1 and according to

Algorithm 2.2

Input: The sorted list of e-sets generated by Algorithm 2.1 corresponding to the given English synset, [*eword1*, *eword2*,... , *ewordn*].

1. Compute the foreign synset as having the following form:

$\{meaning1i_1\} \cup \{meaning2i_2\} \cup \ldots \cup \{meaningni_n\}, \quad 1 \leq i_j \leq m_j, \quad \forall\, j = 1,n$

2. Delete words occurring in more than one e-set from this union, such that each word will occur just once.

Output: The foreign synset corresponding to the given English synset.

It has now become obvious that our approach to WN generation belongs to the class of semiautomatic methods based on heuristics. As it is well known, Atserias *et al.* (1997), such heuristics can belong to two main categories: one in

which the corresponding heuristics rely on information found in the bilingual dictionaries and the structure of WN, another containing heuristics that rely on the genus information extracted from the monolingual dictionary. Obviously, the heuristic which is used here belongs to the first mentioned category, since our generation method does not use monolingual resources (with the exception of WN itself) but relies solely on bilingual dictionaries (in electronic format).

An example

English synset: {personification, incarnation}
the act of attributing human characteristics to abstract ideas etc.

e-sets:

eword	e-set	score
incarnation	{personificare}	2.0
incarnation	{încarnare, întruchipare, întrupare}	1.3333334
personification	{personificare, întruchipare}	1.0

proposed Romanian synset(s):

- {personificare, întruchipare}

3. NOUN SYNSETS

Algorithms 2.1 and 2.2 have been implemented in Prolog and tested by us, with fairly good results, in the case of *Romanian nouns*. In order to test the algorithms, we have used fragments of bilingual dictionaries in electronic format. When working with a semantic network like WN the richness of the bilingual dictionaries which are used is of the essence. Due to the imperfection of existing Romanian-English and English-Romanian dictionaries in electronic format (see, for instance, www.castingsnet.com/dictionaries), and in order to ensure the most possible accurate testing, we have generated our own fragments of electronic bilingual dictionaries, using some of the most complete existing paper ones Levitchi (1973), Levitchi *et al.* (1974). The compiled Romanian-English and English-Romanian dictionaries used in our tests can be seen at

http://phobos.cs.unibuc.ro/roric/wn/r_e.dict

and

http://phobos.cs.unibuc.ro/roric/wn/e_r.dict

respectively. We have randomly chosen a number of 200 English noun synsets for which we have automatically generated the corresponding Romanian ones. Since most English synsets contain two words, our data sample was chosen according to

the same pattern. Thus, out of the 200 considered English synsets, 179 contained two English nouns, 4 synsets contained 3 English nouns and 17 synsets contained more than 3 English nouns (between 4 and 7 words). The number of e-sets involved in the experiment was of 616. Several English synsets containing just one noun have been subsequently taken into consideration. All tests performed have been using the original WN 2.0 in its Prolog-readable format.

The generated Romanian synsets were validated by Romanian linguists using the latest bilingual dictionaries and the corresponding gloss indicated in the American WordNet.

When testing the translation algorithm relatively to Romanian nouns, we have noticed that, in several cases, Algorithm 2.2 has generated more than one Romanian synset corresponding to the given English one. This was the case when Algorithm 2.1 had as output a list of e-sets (corresponding to different meanings of the same word) that had been evaluated with the same value. Each such e-set then represented a candidate and led to a different Romanian or, in general, foreign synset. In such cases the correct foreign synset will be chosen from the list of synsets generated by Algorithm 2.2 according to the gloss of the given English synset. The computer program implementing Algorithm 2.2 must therefore provide as output the gloss as well, since it is necessary in the validation performed by linguists.

When performing tests for Romanian nouns it turned out that, besides the cases when the result was correct, in most other cases the algorithm had generated several Romanian synsets, among which the correct one could be found. In those cases when the English synsets did not have correct Romanian counterparts it was mostly because of wrong or missing data in the bilingual dictionaries. Special problems occurred in the case of English synsets containing a single polysemous noun. For more detailed comments concerning the obtained output see §5.

In order to facilitate the experiment, when choosing our sample of English synsets a necessary step was that of removing the synsets with proper names, compounds and collocations. These should be dealt with separately and with a more significant contribution on the part of the linguists. However, we consider the presented algorithms sufficient for building *a core* of synsets corresponding to all four parts of speech in more or less any language other than English, provided that good bilingual dictionaries in electronic format exist for the specific foreign language involved.

As it is noted in Nikolov, Petrova (2001), the greatest advantage of Algorithm 2.1 is the ability to create synsets which may include foreign words that would not be extracted from the input resource at the first step of the work. Thus, even if a foreign word occurs in the English-Romanian dictionary, for instance, but is missing from the Romanian-English one, there is still a big chance for this word to be included in the final resulting synset. (The only necessary condition for this is the presence in the list of candidates of an e-set which includes that word). This is a

very important fact considering how incomplete bilingual dictionaries usually are. This algorithm, therefore, *does not represent a simple mirror translation*.

Obviously, when using Algorithms 2.1 and 2.2 for specific languages, various difficulties will occur according to what is typical of each language at morphological and derivational level. When testing a variant of the presented translation algorithm for *Bulgarian noun synsets*, for instance, phenomena like the lack of a regular conversion in Bulgarian, the translation of a gerund by a deverbial noun or by a special type of an infinitive or a subordinate clause, the existence of rich systems of participles and others are taken into account in Nikolov, Petrova (2001).

A general difficulty, of a different nature, encountered no matter what language is taken into consideration, consists of what we might call the cross-language wide meaning of a given word. Namely, a word in one language sometimes covers a relatively wide concept and is connected to more than one word in another language, where each of the words it is linked to describes a more specific concept. This is a very important issue from the WN approach point of view, since in WN synsets exist according to the corresponding underlying concepts.

In the case of the Romanian language, we have come to the conclusion that, in those, more interesting cases, in which the bilingual dictionaries are not to blame, the main difficulties that occur when automatically translating the English synsets into Romanian ones were generated by loan translation and by the fact that the polysemy of many English words is greatly superior to that of the corresponding Romanian words (for more details, see §5).

A special case is that of English synsets containing a single polysemous word, a situation in which Algorithm 2.1, or any other algorithm of the same type, will not be able to distinguish among various meanings. This type of difficulty has suggested to us the enrichment technique which is proposed in §4.1 with respect to adjectives, as well as other possible enrichments which will be presented in what follows. Such synset enrichments will take into account the WN structure, proving the necessity of combining class methods with structural ones.

In spite of such difficulties, however, we consider the presented translation algorithms as being appropriate for performing a semiautomatic extraction of the *core* of a foreign WN from the original WN. The most important issue here is the fact that Algorithms 2.1 and 2.2 do not depend on the type of part of speech involved in the translation. It is therefore natural to expect similar or even better results than the ones obtained for nouns when testing with regard to other parts of speech, such as the adjective. Especially since adjectives are less polysemous than nouns.

In what follows, we shall establish how this general algorithm must be enriched in order for it to perform the semiautomatic generation of **adjective synsets and clusters** in languages other than English.

4. ADJECTIVE SYNSETS AND CLUSTERS

WN divides adjectives into two major classes: *descriptive* and *relational*. Chromatic *color adjectives* are regarded as a special case.

In what follows, we shall be concerned with the semiautomatic generation of *adjective clusters* in languages other than English, and will therefore refer solely to descriptive adjectives, which can be organized as this type of structure. The translation of English adjective clusters is completely ensured by the translation of the English adjective synsets and by that of the **ant** relation (denoting antonyms).

4.1. Semiautomatic generation of adjective synsets

In order to translate English adjective synsets into a foreign language Algorithms 2.1 and 2.2 have been used. When translating from English to any other language the *id* which is associated to a synset is not modified. This means that the similarity relation existing between two English synsets will be maintained after performing the translation and will occur among the foreign language adjective synsets as well.

A special problem is posed by synsets containing a single polysemous word. In this case it is impossible to tell which meaning of the word was involved in the creation of the specific synset if one has access to no additional information. The meaning can be guessed only from the gloss. However, in such cases, we have used a strategy which consists in enriching the given synset with new adjectives that suggest the meaning of the one occurring in this synset. The new adjectives are obtained using the similarity relation that typically exists in WN among adjective synsets. Thus, in order to enrich the given synset with new words, the adjectives occurring on the first position within synsets semantically linked to the original one via the similarity relation have been chosen. These words have been appended to the original synset, starting from the second position. This idea was inspired by the way in which adjective clusters are organized and structured in WN. At this point one therefore feels the need to combine the presented class method with a structural one (namely one that takes profit of the WN structure).

The necessary list of e-sets in connection with the given English synset will be generated using Algorithm 2.1. When creating the foreign adjective synset representing the translation of the given English one, Algorithm 2.2 will combine all maximally evaluated e-sets corresponding to each of the words occurring in the English synset. In those cases when more than one e-set will be maximally evaluated corresponding to the same English word, Algorithm 2.2 will generate more than one foreign synset. The final decision concerning the correct translation is then again made according to the gloss.

In order to illustrate how Algorithms 2.1 and 2.2 work in the case of adjective synsets, let us consider the English synset having the *synset_id*[2] 302461205 and containing the unique adjective *sticky*. We shall perform the translation to Romanian of this synset. Let us first note that the chosen target language is not essential for the point that we are trying to make here. The commented results are the output of various Prolog programs which implement the mentioned algorithms.

Since the given English synset contains only one word, it will be enriched as mentioned, according to the similarity relation. After searching the database one comes to the conclusion that the only similarity relation (denoted by the **sim** operator) is

sim(302461205, 302457687).

as well as its symmetrical relation. The synset having *id* = 302457687, which is considered similar in meaning to the one under investigation, contains the unique adjective *wet*. The given English synset is therefore enriched with this adjective. The evaluated e-sets obtained corresponding to the enriched synset, when using the evaluation function mentioned in §2 for Algorithm 2.1, are the following:

evset (302461205, sticky, 1.0, [lipicios, cleios, vascos]).
evset (302461205, sticky, 1.0, [umed, ceţos]).
evset (302461205, wet, 1.0, [umed, jilav, ud]).
evset (302461205, wet, 0.6666666666666666, [ploios, umed, igrasios]).

Here *evset* is an operator designating evaluated e-sets. The first field represents the synset *id*, the second is the ASCII text of the word as entered by the lexicographer, the third gives the value of the evaluating function and the last denotes the foreign evaluated set. In this case the computer program implementing Algorithm 2.2 has the following output:

English synset: [sticky]
Gloss: (moist as with undried perspiration and with clothing sticking to the body; "felt sticky and chilly at the same time")
Romanian synset: [[lipicios,cleios,vascos,umed,jilav,ud],[umed,cetos,jilav,ud]]

One notices that two possible Romanian synsets have been generated. However, only one of them corresponds to the meaning of *sticky* which refers to the

[2] A *synset_id* is a nine byte field in which the first byte defines the syntactic category of the synset and the remaining eight bytes are a *synset_offset*, indicating the byte offset in the file that corresponds to the syntactic category. In the Prolog version of the WN database, the *synset_id*s are used as unique synset identifiers. Also in the Prolog version of WN semantic relations are represented by a pair of *synset_id*s, in which the first *synset_id* is generally the source of the relation and the second is the target.

underlying concept of the synset having *id* = 302461205. The correct foreign (in this case Romanian) synset can be easily chosen according to the corresponding gloss.

Such enrichment with additional words coming from synsets related via similarity with the original one is not always necessary. However, when performed, the chances of empty foreign synsets being obtained (due to the generation uniquely of unlabeled e-sets) are considerably reduced. This operation might produce a slight shift in meaning with respect to the underlying concept of the original English synset. However, only similar concepts are denoted by the involved relation, typical for descriptive adjectives, a fact which determines us to recommend the described strategy. Both translation with and without enrichment can be performed relatively to the same synset (see the following examples), giving linguists the opportunity to compare and to choose among the proposed foreign synsets, when equally taking into consideration the corresponding gloss.

Further examples:

Sample of Automatically Generated Romanian Adjective Synsets (whithout enrichment)

English synset: {clear}

(meteorology) free from clouds or mist or haze; "on a clear day"

e-sets:

eword	e-set	score
clear	{clar, curat, luminos, limpede, senin}	1.0
clear	{limpede, lămurit, inteligibil, clar, desluşit}	1.0
clear	{clar, perceptibil, lămurit, limpede, desluşit}	0.8
clear	{liber, deschis}	0.5
clear	{clar, pătrunzător}	0.5
clear	{curat, neîncărcat, negrevat, întreg}	0.25

proposed foreign synset(s):

- {clar, curat, luminos, limpede, senin}
- {limpede, lămurit, inteligibil, clar, desluşit}

English synset: {fair}

free of clouds or rain; “today will be fair and warm” .

e-sets:

eword	e-set	score
fair	{bun, frumos}	1.0
fair	{ieftin}	1.0
fair	{bălan, bălai, blond, deschis}	0.75
fair	{frumos, curat, îngrijit}	0.6666667
fair	{drept, nepărtinitor, imparţial}	0.6666667
fair	{bun, frumos, plăcut, prielnic, favorabil}	0.6
fair	{cinstit, onest}	0.5
fair	{cinstit, deschis}	0.5
fair	{convenabil, acceptabil, accesibil, rezonabil}	0.5
fair	{bun, natural, firesc}	0.33333334
fair	{frumos, minunat, atrăgător, drăguţ}	0.25

proposed foreign synset(s):

- {bun, frumos}
- {ieftin}

Sample of Automatically Generated Romanian Adjective Synsets (with enrichment)

English synset: {clear}

(meteorology) free from clouds or mist or haze; “on a clear day”

e-sets:

eword	e-set	score
fair	{limpede}	8.0
bright	{senin}	6.0
fair	{senin}	6.0
serene	{clar, senin, limpede}	5.6666665
cloudless	{senin, clar}	5.5

clear	{clar, curat, luminos, limpede, senin}	5.0
bright	{luminos}	5.0
bright	{clar, limpede, transparent}	4.0
clear	{limpede, lămurit, inteligibil, clar, desluşit}	3.0
fair	{citeţ, clar}	3.0
clear	{clar, perceptibil, lămurit, limpede, desluşit}	2.8
clear	{clar, pătrunzător}	2.5
bright	{clar, sonor, cristalin}	1.6666666
serene	{calm, senin, liniştit, potolit, netulburat}	1.4
fair	{frumos, curat, îngrijit}	1.3333334
bright	{spiritual}	1.0
fair	{bun, frumos}	1.0
fair	{ieftin}	1.0
serene	{linştit, calm}	1.0
clear	{curat, neîncărcat, negrevat, întreg}	0.75
fair	{bălan, bălai, blond, deschis}	0.75
fair	{drept, nepărtinitor, imparţial}	0.6666667
fair	{bun, frumos, plăcut, prielnic, favorabil}	0.6
clear	{liber, deschis}	0.5
fair	{cinstit, onest}	0.5
fair	{cinstit, deschis}	0.5
fair	{convenabil, acceptabil, accesibil, rezonabil}	0.5
bright	{inteligent, ager, deştept, sclipitor, scânteietor}	0.4
bright	{strălucitor, scânteietor, sclipitor}	0.33333334
fair	{bun, natural, firesc}	0.33333334
fair	{frumos, minunat, atrăgător, drăguţ}	0.25

proposed foreign synset(s):

- {clar, curat, luminos, limpede, senin}

English synset: {fair}

free of clouds or rain; “today will be fair and warm”

e-sets:

eword	e-set	score
fair	{senin}	2.0
fair	{limpede}	2.0
clear	{clar, curat, luminos, limpede, senin}	1.4
fair	{frumos, curat, îngrijit}	1.3333334
fair	{citeț, clar}	1.0
fair	{bun, frumos}	1.0
fair	{ieftin}	1.0
clear	{limpede, lămurit, inteligibil, clar, desluşit}	1.0
clear	{clar, perceptibil, lămurit, limpede, desluşit}	0.8
fair	{bălan, bălai, blond, deschis}	0.75
clear	{curat, neîncărcat, negrevat, întreg}	0.75
fair	{drept, nepărtinitor, imparțial}	0.6666667
fair	{bun, frumos, plăcut, prielnic, favorabil}	0.6
fair	{cinstit, onest}	0.5
fair	{cinstit, deschis}	0.5
fair	{convenabil, acceptabil, accesibil, rezonabil}	0.5
clear	{liber, deschis}	0.5
clear	{clar, pătrunzător}	0.5
fair	{bun, natural, firesc}	0.33333334
fair	{frumos, minunat, atragător, drăguț}	0.25

proposed foreign synset(s):

- {senin, clar, curat, luminos, limpede}

When studying the above examples, one notices the following: in the case of synset **{clear}**, referring to the meaning coming from meteorology which is given by the mentioned gloss, translation without enrichment leads to two Romanian synsets, corresponding to completely different meanings, out of which the first one represents the correct result (according to the gloss). Performing the same translation with enrichment leads directly to the unique, correct result. In the case

of synset **{fair}**, also referring to the meaning coming from meteorology, translation without enrichment produces two Romanian synsets, which again correspond to completely different meanings, with neither of them representing a correct result. When using the enrichment technique however, a unique, correct Romanian synset is generated.

Performing automatic translation with enrichment will however not always solve all problems, as the previous examples might suggest, and linguistic validation of the obtained output will always be necessary, as can be seen in the following example, where the first generated Romanian synset is the correct one:

English synset: {serene}

completely clear and fine; "serene skies and a bright blue sea"

e-sets:

eword	e-set	score
serene	{clar, senin, limpede}	1.0
serene	{liniştit, calm}	1.0
serene	{calm, senin, liniştit, potolit, netulburat}	0.6

proposed foreign synset(s):

- {clar, senin, limpede}
- {liniştit, calm}

(In this particular case, using the existing bilingual dictionaries, it turned out that translation with and without enrichment lead to the same result).

4.2. Semiautomatic generation of adjective clusters

The translation of English adjective clusters is completely ensured by the translation of the English adjective synsets and by that of the **ant** relation (denoting antonyms). Since the translation of adjective synsets has already been discussed, let us now refer to the translation of the **ant** relation. This becomes a very important issue when taking into account the fact that antonym dictionaries in electronic format do not exist for a great number of languages.

In the Prolog version of the WN database, which we have been using here, semantic relations are represented by a pair of *synset_id*s, in which the first *id* is generally the source of the relation and the second is the target, as is the case with

the already mentioned **sim** operator. If two pairs *synset_id*, *w_num* are present, the operator represents a lexical relation between word forms, where *w_num* specifies the word number for a specific word in a specific synset. If present, *w_num* indicates which word in the synset is being referred to. The **ant** operator, for instance, specifies antonymous words in the following form:

ant(*synset_id,w_num,synset_id,w_num*).

Thus, the significance of the following Prolog fact

ant(302425368, 1, 302423307, 1).

is that the first word of the synset having the *id* 302425368 and the first word of the synset having the *id* 302423307 are *direct antonyms*. This is a lexical relation that holds for all syntactic categories but is essential in the formation of adjective clusters. For each antonymous pair, both relations are listed (i.e. each *synset_id,w_num* pair is both a source and a target word).

When studying the contents of file **wn_ant.pl** of the WN Prolog database, which contains all Prolog facts referring to antonymous words, one easily notices that the great majority of these facts establish direct antonymy relations among words occurring as first elements within the synsets to which they belong. The exceptions can be easily processed by a human operator retaining the new positions of the adjectives having direct antonyms. Under these circumstances, we have found it justifiable to formulate

Remark 4.1

The first word of an English adjective synset is the one possibly having a direct antonym.

Let us now assume that all (translated) foreign adjective synsets exist, corresponding to a given language, and that they belong to a file named **wn_strans.pl**. Using Remark 4.1 and having generated file **wn_strans.pl** by applying the translation algorithm, we can now formulate the algorithm for generating the foreign adjective clusters corresponding to the English ones:

Algorithm 4.1

Input: Files **wn_ant.pl**, **wn_sim.pl**, and **wn_strans.pl**

For each *synset pair* denoted by each Prolog fact of file **wn_ant.pl**, perform steps 1. through 5.:

1. Look in file **wn_strans.pl** and find the foreign synsets representing the translations of the considered English ones.
2. Corresponding to each foreign synset found in **wn_strans.pl** in step 1. retain the first word of that synset. (This word pair will be used in the foreign cluster head).

3. For the same word pair look in file **wn_sim.pl** and take into consideration the **sim** clauses corresponding to each of the two synsets to which the two words of the cluster head belong.
4. Take into account all synsets denoted by the **sim** clauses chosen in step 3., synsets having the second *id* which occurs in the clause. Find the foreign synsets representing their translations in file **wn_strans.pl**.
5. Add each first word of these foreign synsets in the cluster head, together with the & pointer.
6. Add each "similar" foreign synset, ending it with the reciprocal similarity pointer.

Output: A file containing all foreign adjective clusters.

Algorithm 4.1 will generate foreign adjective clusters with a bipolar structure like the one described in [Miller et. al., 90].

At this early stage of our study we have been concerned uniquely with creating the WN *type* cluster structure and have not tried to distinguish among different subsenses or different privileges of occurrence. We have equally not tried to indicate the limitation of certain adjectives as to the syntactic positions they can occupy, a word-form limitation which in WN is coded for individual adjectives. This can easily be achieved once the basic algorithm has been established. Other issues, such as the capitalized pointers sometimes occurring in the structure of WN clusters, which serve as "see also" cross-references to related clusters, have also been ignored for the time being. All these and others represent topics for future study.

Obviously, according to the chosen target language, various difficulties of linguistic nature will be encountered. For instance, identical foreign synsets might be generated by Algorithm 4.1 corresponding to different English ones, namely to different meanings and concepts. This is the case when an English polysemous adjective will have one or more meanings in English that do not exist in the target language, a situation which is called *semantic loan*, leading to *loan translation*. Linguistic validation of the output of computer programs implementing Algorithm 4.1, or any other algorithm of the same type, for that matter, will always be necessary. However, we consider that Algorithm 4.1 accounts for the great majority of cases when dealing with adjective clusters of WN type.

4.3. Final remarks concerning adjectives

The proposed approach to WN generation is a combination of automatic and manual methods. The manual method relies on human experts, while the automatic class method relies uniquely on bilingual dictionaries.

Using the proposed class method (which is language independent and irrespective of part of speech) is sufficient in order to automatically generate the synsets of the target language (which will be manually validated). In the case of

adjectives, however, one should be concerned not only with automatically translating English adjective synsets but also with creating the typical adjective cluster structure corresponding to the target language. In order to achieve this, the WN structure should be taken into account, a fact which denotes the necessity of combining class methods with structural ones. The generation of adjective clusters can be accomplished entirely automatically (using Algorithm 4.1), provided that the translation of the involved adjective synsets, performed by means of the proposed class method, has been validated by human experts.

The significance of the manual effort involved in quality assurance primarily depends on the existence of appropriate tools. The involved human effort can be greatly reduced in the case of those languages for which correct and complete bilingual dictionaries in electronic format exist.

5. MORE LINGUISTIC COMMENTS CONCERNING THE OBTAINED OUTPUT

It is our belief that one should start any linguistic comments concerning this type of output by mentioning, from the very beginning, that, in most cases when the obtained results are not the best possible ones, it is mainly because of the imperfection of existing bilingual dictionaries.

In Hristea, Th. (2003: 153), to which we shall refer in what follows, three types of situations are considered the most interesting: “those in which the program has generated more than one Romanian synset, out of which one is correct, those in which no Romanian synset has been generated, and those, very rare cases, in which one or more synsets have been generated, none of them being correct”.

In the first case choosing the correct translation can easily be performed by linguists (according to the corresponding gloss), while in the second, more interesting case, it is usually the bilingual dictionaries that are to blame. “Sometimes only one of the dictionaries is to blame, usually the Romanian-English one, relatively poor concerning the number of entries, but also as far as the number of English words taken into consideration for performing translations is concerned. Due to this fact, there are many cases in which only unlabeled e-sets are obtained via the proposed algorithm. No Romanian synset will be generated in such cases” (Hristea, Th. 2003: 153). Among the situations in which a Romanian word occurring in the English-Romanian dictionary is not found in the Romanian-English one, the following one is underlined: “it is especially the case of nouns coming from verbs and having the significance <<the action of...>>. Important and frequent Romanian words like **organizare** (coming from <<a organiza>> – <<to organize>>) or **respingere** (coming from <<a respinge>> - <<to reject>>), occur as translations of various English words but are not to be found in the Romanian-English dictionary. This can determine the algorithm for the evaluation of e-sets to fail, since the absence of a word from the Romanian-English dictionary leads to a lower value of the corresponding e-set” (Hristea, Th. 2003: 154).

Hristea, Th. (2003: 154) additionally notes that "also due to the incompleteness of existing bilingual dictionaries many recent borrowings which exist in Romanian (especially in mass-media) will not occur in the generated Romanian synsets".

As the same author notices Hristea, Th. (2003: 154), "sometimes the Romanian synset generated by the program is incorrect because of the evaluation function which was implemented. Other evaluation functions should be implemented and tested in future studies".

However, the most interesting cases are those in which the dictionaries, in general, and the Romanian-English one in particular, are not to blame: "in those, more interesting, cases in which the Romanian-English dictionary is not to blame, the cause of the errors which the programs generate is of a completely different nature. One should look for it in connection with concepts. In this case one must take into account the fact that English in general and American English, to which WordNet refers, in particular, is a much richer language than Romanian. Statistically speaking, while Romanian has a maximum of 150,000 words, American English includes approximately 450,000 words (according to information provided by the lexicographer St. Berg Flexner). But, in comparison with Romanian, English is a much more advanced language not only from a grammatical and lexical point of view. Quantitatively it includes more words or lexical units. However, English is much more advanced from the semantic point of view as well, since an English word often has a much richer semantic content than the corresponding Romanian one. Numerous words existing both in English and in Romanian are more polysemous in English than in Romanian. In other words, the polysemy of many English words is greatly superior to that of the corresponding Romanian ones. For instance, the English word **feature** having the meaning of <<an article of merchandise that is displayed or advertised more than other articles>> has no correspondent in Romanian. No single word with this meaning exists. We are therefore obliged to perform translation using a group of words (a gloss), while the English synset containing the sole word **feature** which refers to this concept will have no Romanian counterpart. In this case the computer program did not work correctly. It is, once again, a situation which affects primarily English synsets containing a single word. Another example of an English polysemous word is **foundation**, which attracted our attention through one of its meanings, that of <<a woman's undergarment worn to give shape to the contours of the body>>. This meaning of **foundation** does not exist in Romanian. The concept to which the synset containing the unique word **foundation** with this meaning refers to should be explained in Romanian by means of a gloss. No corresponding Romanian synset should exist. The computer program has again failed in this case, just as it has in the case of the English **quiver** having the meaning <<a case for holding arrows>>" (Hristea, Th. 2003: 154-155).

Another situation in which the program did not work correctly is, according to the linguist Hristea, Th. (2003: 155), that in which specific English nouns are used with a negation. "This is, for instance, the case of **matter** with negation, as in

<<they were friends and it was no matter who won the game>>. This English noun should be translated into Romanian by a collocation, centered around a noun which does not occur in the English-Romanian dictionary among the possible translations of **matter**. Another possibility is that it does occur, however by means of an equivalent of collocational type, that will not be used by the algorithm which the program implements. In such cases the program can not determine the Romanian (or, in general, the foreign) synset correctly. Specifically, in the case of **matter** used with a negation, several possible Romanian synsets have been generated. None of them is, however, correct, since none of them includes the noun **importanță** (importance), which occurs in the Romanian collocation corresponding to this meaning. This Romanian collocation represents a loan translation of the French <<avoir de l'importance>>. Loan translations after French are extremely frequent in Romanian. This is why we feel the need for future programs to take into account collocations, both in English and in Romanian, or, more generally, in the target language".

As the same linguist notes, [Hristea, Th., 03], "another source of difficulties was represented by nouns in plural form. Some of the English synsets contain nouns in singular form which should be translated by plurals in Romanian. Examples from this category are **foundation** translated by the plural **fonduri**, or **knowledge** translated by **cunoştinţe"**. This comment has determined us to include the plural forms of such nouns in the Romanian-English dictionary that should be used by the computer program implementing Algorithm 2.1.

The important relationship between homonymy and polysemy is equally taken into account Hristea, Th. (2003: 156): "In Romanian, as in other languages, like French, for instance, the relationship between homonymy and polysemy represents an extremely complicated issue, a problem which is not yet solved. In many cases, according to various researchers, one deals with two, three or even more homonymous words, while according to others with a unique polysemous word, having two, three or even more fundamental meanings, which are more or less related to one other. An example would be the word **bun** (**good**), which in Romanian is primarily an adjective having seven fundamental meanings. Secondly it represents a noun having two different plural forms, which are semantically specialized. The Romanian noun **bun** (**good**) having the plural **bunuri** has four meanings, while the same noun **bun** with plural form **buni** has only one meaning, that of grandfather. These situations occur quite frequently in Romanian". In order for the computer programs which implement the described algorithms to produce better results, the author recommends using "dictionaries which treat possible homonyms, especially the so-called semantic ones, as a single polysemous word" (Hristea, Th. 2003: 156)..

Although the mentioned author Hristea, Th. (2003) is concerned primarily with mistakes occurring when performing automatic translation, he concludes by noticing that "in most cases when the bilingual dictionaries were correct and

complete, the implemented algorithm proved to work surprisingly well. Thus, in the case of concepts which are very close to one another in English, the existing subtle difference in meaning has been sensed by the algorithm which correctly maintains it in the Romanian translation. It is, for instance, the case of the English synsets **[banishment, proscription]** having the meaning <<the act of banishing someone>> and **[ostracism]** having the meaning <<the act of excluding someone from society by general consent>> respectively. The first was translated into Romanian by the synset **[exilare, surghiunire, exil, surghiun, expulzare, ostracizare]**, while the second one was translated into the unique **[ostracism]**. The Romanian **ostracism** is the only of all these synonym words which also refers to consensus in making the banishment decision. Its occurrence in the second synset, as a unique element, points out the subtle difference between the two concepts to which the English synsets refer" (Hristea, Th. 2003: 157).

The same linguist Hristea, Th. (2003) concludes by noticing that the main difficulties which occurred when automatically translating English synsets into Romanian ones were generated by collocations, by loan translation, and by the fact that the polysemy of many English words is greatly superior to that of the corresponding Romanian words. Additionally, he points out that "most problems occurred when translating English synsets that contain a single word, the algorithm often being unable to decide among meanings. Such synsets should probably be subject to further investigation. On the other hand, we would like to emphasize the fact that, in the absence of truly competitive tools (with reference to paper and electronic dictionaries) the realistic evaluation of the computer programs becomes rather difficult, if not almost impossible" (Hristea, Th. 2003: 156-157).

6. NOUN SYNSETS REVISITED

Having as starting point the linguist's remark Hristea (2003) that "most problems occurred when translating English synsets that contain a single word, the algorithm often being unable to decide among meanings", we now revisit the English synsets containing a unique polysemous noun.

It again becomes obvious that, in order to correctly distinguish among the existing meanings, when performing automatic translation, the proposed algorithm should take into account the WN structure as well. However, in the case of noun synsets, the similarity relation which has been used by us with respect to adjective ones does not exist. This leads us to taking into consideration a completely different type of relation, that of **hypernymy**, as suggested in Sima and Vaţă (2004). Hypernymy is one of the basic semantic relations implemented in WN, which corresponds to the **isa** relation and according to which nouns are structured as hierarchies. When automatically translating noun synsets containing a unique polysemous noun, we have therefore directed our attention towards the involved

synset's hyperonym, denoting the mother concept of the one that the synset under investigation refers to.

In the Prolog version of the WN database, that we have been using, the hypernymy relation is expressed under the following form:

hyp(*synset_id*, *synset_id*).

The **hyp** operator specifies that the second synset is a hypernym of the first synset. This relation holds for nouns and verbs. The reflexive operator, hyponym, implies that the first synset is a hyponym of the second one.

We have been trying to estimate to what extent synset enrichment performed by means of hypernyms increases the chances of correctly translating English synsets that contain a unique polysemous noun. The total number of such synsets existing in the 1.7.1. version of WN, which has been used here in all experiments, is 13448[3]. Taking this figure and these synsets into account as input, a random selection has been generated, as follows Sima and Vaţă (2004), Hristea and Vaţă (2005):

1. We have randomly selected (using the Rand( ) function which is implemented in Perl 5.8.0) a number of 135 synsets containing a unique polysemous noun. These selected synsets represent 1/100 of the total input.
2. We have added to this data sample all other synsets having the same contents as those generated at the previous step. (*Example*: If, in step 1, the synset [bearing], having the *synset_id* 102450394, has been selected, then, in step 2, all synsets containing the unique polysemous noun *bearing* will be included in the data sample. In this case only one such synset exists, namely the one having the *synset_id* 111640712). After performing step 2 of this simulation, we have obtained a number of 257 noun synsets which will be used in our estimation and which represent 1.91% of the original input.
3. Each of the 257 noun synsets obtained in step 2 has been enriched by adding all nouns of the first hypernym[4] synset.

Remark 6.1

The choice of the "first hypernym" as being the most significant to be used for performing enrichment has been made according to the following simulation: a number of 199 synsets containing a unique polysemous noun and having multiple hypernyms exists. (This represents only 1.4% of the total number of synsets under investigation). We have randomly selected 20 such synsets (representing

[3] Since the previous version 1.7.1 of WN contains a larger number of synsets being formed with a unique polysemous noun than version 2.0, the former has been chosen for performing the tests which led to the formulation of Algorithm 6.1.

[4] Here "first hypernym" refers to the ordering existing among *synset_id*s.

approximately 10% of the synsets having multiple hypernyms) and have come to the conclusion that, in 17 cases out of 20 (namely in 85% of all cases), one can retain only the first hypernym, which can be considered the most significant.

Algorithm 2.1 has been used for performing the automatic translation to Romanian of all synsets containing a unique polysemous noun, as well as of the 257 enriched synsets obtained as previously described. In the first case (non-enriched synsets), the algorithm has failed to produce a correct translation in 60.75% cases. When dealing with enriched synsets, the same algorithm has failed in only 31% cases. This shows us that the proposed enrichment technique decreases failure in automatic unique polysemous noun synset generation with approximately 50%, a result which encourages us to reformulate Algorithm 2.1, corresponding to this part of speech, as follows Sima and Vaţă (2004), Hristea and Vaţă (2005):

Algorithm 6.1

Input: The English synset which is to be translated, the file **wn_s.pl** (where an **s** operator is present for every word sense in WN), the file corresponding to the **hyp** operator, and the two files representing the two bilingual dictionaries.

1. If the given English synset consists of just one noun, find out (by consulting the **wn_s.pl** WN file) if this noun is a polysemous one. If not, STOP and use Algorithm 2.1 for performing translation.
2. Use the **hyp** operator file in order to find the first hypernym of the given synset. If such a hypernym exists[5], do:
   - **2.1.** Enrich the given English synset containing a unique polysemous noun with all nouns of the synset representing its first hypernym.
   - **2.2.** Delete one of the occurrences of the noun of the original synset if this word also exists in the synset used to perform the enrichment. The newly resulting (enriched) synset is the one to be translated.
3. Create (by consulting the appropriate bilingual dictionary) the e-sets corresponding to each word of the English synset to be translated.
4. Label the English words belonging to the given English synset.
5. Label each of the e-sets generated in step 2.
6. Remove all unlabeled e-sets.
7. Evaluate the e-sets (using the assigned labels and an evaluating function).

[5] In WN all noun synsets have hypernyms. An exception is represented only by the top level synsets. Among these synsets, there is one containing a unique polysemous noun, namely the synset {state} having as gloss (the way something is with respect to its main attributes; “the current state of knowledge”; “his state of health”; “in a weak financial state”).

8. Sort (according to their scores and to the English words they correspond to), in ascending or descending order, the obtained list of evaluated e-sets.
9. Choose the e-set corresponding to the noun in the original English synset which is evaluated with the highest score and present it as output. STOP.

Output: The foreign synset corresponding to the original English synset.

Remark 6.2

Since the hypernym synset has been used only for *specifying the meaning* of the polysemous noun occurring in the English synset to be translated, no e-sets corresponding to nouns of the hypernym synset will be selected when forming the foreign synset that represents the translation. Using Algorithm 2.2 is therefore not necessary in this case. Algorithm 6.1 has as output the final result, namely the foreign synset representing the translation in the target language of the given English one (see also the following example).

An example

In order to illustrate how Algorithm 6.1 works, let us consider translating into Romanian the three English synsets containing the unique polysemous noun **tiller** and having, in WN 1.7.1, the *synset_id*s 103867107, 108769239 and 111093313 respectively[6]. In what follows, the mentioned results are the output of a Prolog program, as presented in Sima and Vaţă, (2004).

In the case of the English synset **[tiller]** having *id* 103867107 and the corresponding gloss *lever used to turn the rudder on a boat*, the enriched synset resulting in step 2.2 of Algorithm 6.1 is **[tiller,lever]** and the computer program implementing Algorithm 6.1 has the following output:

English synset: [tiller]
(lever used to turn the rudder on a boat)

sin(103867107,[tiller,lever]).
evset(103867107,tiller,1,[bielă]).
evset(103867107,lever,0.75,[parghie,levier,mâner,braţ]).
evset(103867107,lever,1,[mâner]).

evset(103867107,tiller,1,[biela]).

Romanian synset:
[[biela]]

[6] Let us note that version 2.0 of WN includes the same three synsets containing the unique polysemous noun **tiller** and having the *synset-id*s 104264311, 110012569 and 112411086 respectively.

The e-set chosen in step 9 of Algorithm 6.1, in this case, is therefore that given by the Prolog clause

evset(103867107,tiller,1,[bielă]).

and the final result is the Romanian synset **[biela]**, which represents a correct translation.

In the previously presented output *evset* is an operator designating evaluated e-sets. The first field represents the *synset_id*, while the second is the ASCII text of the word as entered by the lexicographer. The numbers corresponding to the third field represent the values of the evaluation function in the case of each e-set obtained in step 3 of Algorithm 6.1. The last field denotes the foreign (in this case Romanian) evaluated e-set. The Prolog predicate *sin* has two arguments, the first one being the *synset_id* of the synset to be translated, while the second represents the (enriched) synset that we are actually translating. *Sin* entries correspond only to those synsets which are to be translated and which are considered "complete" (meaning that entries in the English-Romanian dictionary exist corresponding to all occurring English words).

In the case of the English synset **[tiller]** having *id* 108769239 and the corresponding gloss *someone who tills land (prepares the soil for the planting of crops)*, the enriched synset resulting in step 2.2 of Algorithm 6.1 is

[tiller, sodbuster, granger, husbandman, farmer]

and the computer program implementing Algorithm 6.1 now has the following output:

English synset: [tiller]
(someone who tills land (prepares the soil for the planting of crops))

sin(108769239,[tiller,sodbuster,granger,husbandman,farmer]).
evset(108769239,tiller,2,[agricultor,plugar]).
evset(108769239,tiller,1,[bielă]).
evset(108769239,sodbuster,4,[cultivator]).
evset(108769239,granger,1,[agricultor,gospodar]).
evset(108769239,husbandman,2,[cultivator,agricultor,gospodar]).
evset(108769239,farmer,0.5,[fermier,arendas]).

evset(108769239,tiller,2,[agricultor,plugar]).
evset(108769239,tiller,1,[biela]).

Romanian synset:
[[agricultor,plugar]]

The e-sets which are now generated corresponding to the noun *tiller* of the original English synset are **[agricultor,plugar]** and **[biela]**. The e-set chosen in step 9 of Algorithm 6.1 is the higher evaluated **[agricultor,plugar]**, which denotes the corresponding Romanian synset and which again represents a correct translation of the English given one.

Finally, in the case of the English synset **[tiller]** having *id* 111093313 and the corresponding gloss *a shoot that sprouts from the base of a grass*, the enriched synset resulting in step 2.2 of Algorithm 6.1 is **[tiller,shoot]** and the computer program implementing Algorithm 6.1 has the following output:

English synset:[tiller]
(a shoot that sprouts from the base of a grass)

sin(111093313,[tiller,shoot]).
evset(111093313,tiller,1.33333,[vlăstar,mlădita,puiet]).
evset(111093313,tiller,1,[bielă]).
evset(111093313,shoot,0.666667,[mlădiță,vlastar,mugurel]).
evset(111093313,shoot,0.5,[scoc,jgheab]).

evset(111093313,tiller,1.33333,[vlăstar,mlădiță,puiet]).
evset(111093313,tiller,1,[biela]).

Romanian synset:
[[vlăstar,mlădiță,puiet]]

The generated e-sets corresponding to the noun *tiller* of the original English synset are **[vlăstar,mlădiță,puiet]** and **[bielă]**. In step 9 of Algorithm 6.1 the higher evaluated **[vlăstar,mlădiță,puiet]** is chosen and presented as output. This again represents a correct translation of the given English synset.

Let us now note that the translation of the three English synsets containing the unique polysemous noun *tiller* has also been performed by us using Algorithms 2.1 and 2.2 (namely without enrichment). In all three cases the obtained result was the unique Romanian synset **[bielă]**[7]. Algorithm 2.1 was therefore unable to distinguish among the different existing meanings and the obtained result of the translation can only be correct in one of the three studied cases. These are the typical results to be obtained within this type of framework, and such results again demonstrate the need to combine class methods with structural ones when dealing with the WN semantic network in order to perform automatic translation.

[7] Corresponding to all studied synsets, **[bielă]** was *the only labeled e-set* and therefore represented the only possible choice.

6.1. Final remarks concerning nouns

Just as previous authors Nikolov and Petrova (2001), Hristea (2002), while performing tests for the Romanian language, in order to facilitate the experiment, we have not taken into account synsets with proper names, compounds and collocations. As it is noted in Hristea (2002), "these should be dealt with separately and with a more significant contribution on the part of the linguists."

In order to decrease failure in the automatic translation of English synsets containing a unique polysemous noun, as well as to minimize the involved human effort, we have included an "enrichment step" in the existing translation Algorithm 2.1. In the case of the Romanian language, this improves automatic translation of such noun synsets with approximately 50%.

Our improved method reinforces the necessity of combining class methods with structural ones when dealing with this type of task. It relies strongly on the existence of comprehensive bilingual dictionaries in electronic format. That is why, in the absence of truly competitive tools (with reference to both paper and electronic dictionaries), the realistic evaluation of the involved computer programs becomes rather difficult, if not almost impossible.

7. VERB SYNSETS

In discussing the semiautomatic generation of verb synsets in languages other than English we shall extend the same algorithmic approach suggested in Nikolov and Petrova (2001) and proposed in Hristea (2002), while again stating the need to combine class methods with structural ones for the automatic construction of such lexical data bases.

Whenever the synset to be automatically translated contains multiple verbs, translation will be performed using Algorithms 2.1 and 2.2. A special problem is again posed by synsets containing *a unique polysemous verb* since, in such cases, there is no way of knowing which meaning of the involved word a specific synset refers to.

The proposed translation algorithm Hristea (2003) will be using the same concepts and the similarity in meaning of various verb synsets which, in the Prolog version of the WN database, is expressed under the following form:

vgp (*synset_id*, *synset_id*).

The **vgp** operator specifies verb synsets that are similar in meaning and that should be grouped together when displayed in response to a grouped synset search. This relation only holds for verb synsets.

The proposed algorithm for generating the foreign *e-sets* corresponding to a given English synset which contains a unique verb is the following:

Algorithm 7.1

Input: The English synset which is to be translated, the file **wn_s.pl** (where an **s** operator is present for every word sense in WN), the file corresponding to the **vgp** operator, and the two files representing the two bilingual dictionaries.

1. Find out (by consulting the **wn_s.pl** WN file) if the unique verb of the synset to be translated is a polysemous one. If not, STOP and use Algorithm 2.1 for performing translation.
2. Use the **vgp** operator file in order to find a synset similar in meaning. If such a synset exists, do:
   - **2.1.** Enrich the given English synset containing a unique polysemous verb with an entire synset which is similar in meaning.
   - **2.2.** Delete one of the occurrences of the verb of the original synset if this word also exists in the synset used to perform the enrichment. The newly resulting (enriched) synset is the one to be translated.
3. Create (by consulting the appropriate bilingual dictionary) the e-sets corresponding to each word of the English synset to be translated.
4. Label the English words belonging to the given English synset.
5. Label each of the e-sets generated in step 2.
6. Remove all unlabeled e-sets.
7. Evaluate the e-sets (using the assigned labels and an evaluating function).
8. If the given English synset has been enriched in step 2.1, then process the list of evaluated e-sets obtained in step 7 as follows: if the same e-set occurs identically (i.e. contains the same words and in the same order) corresponding to both the verb of the original English synset and to all other verbs that have been used for enrichment, then add up all obtained scores of this e-set; evaluate this e-set corresponding to the verb of the original English synset using the total score.
9. Sort (according to their scores and to the English words they correspond to), in ascending or descending order, the obtained list of evaluated e-sets.

Output: The sorted list of e-sets corresponding to the given English synset.

Let us note the fact that step 8 of this algorithm represents the main way in which the proposed enrichment technique contributes to specifying the meaning of the unique polysemous verb occurring in the original English synset (see also the example in §7.1 of the present paper).

The sorted list of e-sets generated by Algorithm 7.1 should be further used as input for Algorithm 2.2, which will have as output the foreign synset corresponding to the given English one.

7.1. Examples and final remarks concerning verbs

In order to illustrate how the translation algorithm works, let us consider translating two English verb synsets into the corresponding Romanian ones. In what follows, all presented results are the output of various Prolog programs.

Let us first consider the English synset **[unite, unify, merge]**, having (in WN 2.0) the *synset_id* 200235024 and referring to the concept defined by the following gloss: *join or combine; "We merged our resources"*. The evaluated e-sets obtained corresponding to this synset, in step 5 of Algorithm 2.1, when using the same evaluation function as before, are the following:

evset(200235024,unite,3,[a_uni, a_reuni, a_unifica]).
evset(200235024,unite,1,[a_lega]).
evset(200235024,unify,3,[a_unifica, a_uni, a_reuni]).
evset(200235024,merge,4,[a_contopi, a_fuziona, a_uni]).

Here *evset* is the same operator designating evaluated e-sets as in the previous sections. In this case, the computer program implementing Algorithm 2.2 has the following output:

The translation of synset 200235024 = [unite, unify, merge] having g=(join or combine; "We merged our resources"): trad(200235024, [a_uni, a_reuni, a_unifica, a_contopi, a_fuziona]).

The translation of the given English synset offered by the implemented algorithms, by means of the operator *trad*, is therefore the Romanian verb synset **[a_uni, a_reuni, a_unifica, a_contopi, a_fuziona]**, which represents, according to Romanian linguists, an appropriate translation.

Let us now consider an English synset containing a unique polysemous verb, in order to test the enrichment technique proposed in step 2 of Algorithm 7.1. For this purpose we have randomly chosen the English synset **[change]**, having the *synset_id* 200162972 and referring to the concept defined by the following gloss: *change clothes; put on different clothes; "Change before you go to the opera"*.

Let us first note that the polysemous verb *change* is recorded in version 2.0 of WN as having ten different meanings. It therefore occurs in ten different synsets (not taking collocations into account). In three of these ten synsets *change* is the only word occurring. However, not all of the three synsets can be enriched, as in step 2 of Algorithm 7.1, since other synsets that are similar in meaning do not exist corresponding to all of them.

Getting back to the synset **[change]** having *id* 200162972, after searching the **vgp** operator file at the second step of Algorithm 7.1, one finds out that this synset is similar in meaning to the verb synset having *id* 200535406:

vgp(200162972, 200535406).

The latter is the synset **[switch, shift, change]**, referring to the concept denoted by the following gloss: *lay aside, abandon, or leave for another; "switch to a different brand of beer"; "She switched psychiatrists"; "The car changed lanes".*

The given synset **[change]** will therefore be enriched with this entire new synset obtaining, in step 2.2 of Algorithm 7.1, the synset **[change, switch, shift]**, which is, in fact, the one to be translated.

The evaluated e-sets obtained corresponding to this synset, in step 7 of Algorithm 7.1, when using the same evaluation function, are the following:

evset(200162972, change, 2, [a_schimba, a_modifica]).
evset(200162972, change, 1, [a_altera]).
evset(200162972, change, 2, [a_preface, a_transforma]).
evset(200162972, change, 1, [a_converti]).
evset(200162972, change, 1, [a_schimba]).
evset(200162972, switch, 2, [a_schimba]).
evset(200162972, shift, 2, [a_schimba]).

One notices (in step 8 of Algorithm 7.1) that the same e-set **[a_schimba]** occurs corresponding to both the verb *change* of the original English synset and to all other verbs that have been used for enrichment. After computing the total score of this e-set corresponding to the verb *change*, in step 8 of the algorithm, and after sorting the new list of evaluated e-sets in step 9, one obtains the following *final list of evaluated e-sets*, representing the input for Algorithm 2.2:

evset(200162972, change, 5, [a_schimba]).
evset(200162972, change, 2, [a_schimba, a_modifica]).
evset(200162972, change, 2, [a_preface, a_transforma]).
evset(200162972, change, 1, [a_altera]).
evset(200162972, change, 1, [a_converti]).
evset(200162972, switch, 2, [a_schimba]).
evset(200162972, shift, 2, [a_schimba]).

The computer program implementing Algorithm 2.2 has chosen the correct Romanian synset **[a_schimba]** as the translation of the given English synset **[change]** (having *id* 200162972). The corresponding Romanian synset will be assigned the same *synset_id*.

Let us now see what happens when trying to translate into Romanian the same synset **[change]** having *id* 200162972, but without performing the synset enrichment required in step 2 of Algorithm 7.1.

In this case one obtains the following evaluated e-sets in step 5 of Algorithm 2.1:

evset(200162972, change, 2, [a_schimba, a_modifica]).
evset(200162972, change, 1, [a_altera]).
evset(200162972, change, 2, [a_preface, a_transforma]).
evset(200162972, change, 1, [a_converti]).
evset(200162972, change, 1, [a_schimba]).

The translation of the chosen synset **[change]**, performed without enrichment, points out (Algorithm 2.2) two possible Romanian synsets:

[a_schimba, a_modifica] and **[a_preface, a_transforma]**.

The choice as to which synset represents the correct translation must be made according to the gloss and involves additional human effort. However, one notices that both proposed synsets are inappropriate translations of the given English one, since here the meaning of *change* is neither that of modifying, nor that of transforming.

When studying such examples, the necessity of using the **vgp** operator and corresponding relation, whenever possible, becomes obvious. This is why, unlike other authors who use the concept of e-set, while restricting their approach to class methods Nikolov and Petrova (2001), we strongly believe in the necessity of combining such methods with structural ones.

Just as in the case of nouns and adjectives, a special problem is posed by synsets containing a unique polysemous verb. In such cases, there is no way of knowing which meaning of the involved word a specific synset refers to. The only way of finding out is by checking the associated gloss. In order to minimize the human effort involved and to automate the translation process as much as possible, the enrichment step has been included in Algorithm 2.1. The second step of the newly resulting algorithm uses a semantic relation that is typical of verb synsets and which is of great importance in specifying the concept to which the given English synset refers, as could be seen in the previous example.

8. ADVERB SYNSETS

Since no semantic relation of type "similarity in meaning" is available between adverb synsets, we have performed the translation of such synsets by means of Algorithm 2.1 and Algorithm 2.2, therefore restricting our approach to using a class method. However, in order to compare results, we have implemented the first mentioned algorithm using two different evaluation functions, namely that described in §2 and one obtained in the same way, but without performing division by the size of the e-set.

In what follows, we present two examples of translation. In each case the first table and corresponding result having been obtained when using the evaluation function of §2, while the second was obtained when using the other evaluation function respectively.

Example no. 1

English synset: {happily, merrily, mirthfully, gayly, blithely, jubilantly, with_happiness}

in a joyous manner ; “they shouted happily”

e-sets:

eword	e-set	score
jubilantly	{vesel}	7.0
blithely	{vesel, voios, lipsit_de_griji}	3.6666666666666665
happily	{fericit, vesel, entuziast}	3.0
merrily	{voios, cu_veselie}	3.0
gayly	{vesel, jovial, binedispus}	2.9999999999999996
mirthfully	{cu_veselie, cu_voioşie, cu_bucurie}	2.6666666666666665
with_happiness	{cu_veselie, cu_jovialitate}	2.5
happily	{bucuros, cu_bucurie}	2.0
with_happiness	{cu_noroc, cu_bucurie, cu_fericire}	1.6666666666666665
happily	{din_fericire}	1.0
gayly	{ţipător}	1.0
gayly	{destrăbălat, deşucheat}	1.0
gayly	{fără_cumpătare}	1.0
jubilantly	{triumfător, jubilând}	1.0
with_happiness	{cu_fericire, cu_mulţumire, cu_satisfacţie}	1.0
gayly	{zgomotos, ţipător}	0.5

proposed foreign synset(s):

- {vesel, voios, lipsit_de_griji, jovial, binedispus, fericit, entuziast, cu_veselie, cu_voioşie, cu_bucurie, cu_jovialitate}

English synset: {happily, merrily, mirthfully, gayly, blithely, jubilantly, with_happiness}

in a joyous manner ; “they shouted happily”

e-sets:

eword	e-set	score
blithely	{vesel, voios, lipsit_de_griji}	11.0
happily	{fericit, vesel, entuziast}	9.0
gayly	{vesel, jovial, binedispus}	9.0
mirthfully	{cu_veselie, cu_voioşie, cu_bucurie}	8.0
jubilantly	{vesel}	7.0
merrily	{voios, cu_veselie}	6.0
with_happiness	{cu_noroc, cu_bucurie, cu_fericire}	5.0
with_happiness	{cu_veselie, cu_jovialitate}	5.0
happily	{bucuros, cu_bucurie}	4.0
with_happiness	{cu_fericire, cu_mulţumire, cu_satisfacţie}	3.0
gayly	{destrăbălat, deşucheat}	2.0
jubilantly	{triumfător, jubilând}	2.0
happily	{din_fericire}	1.0
gayly	{fără_cumpătare}	1.0
gayly	{ţipător}	1.0
gayly	{zgomotos, tipător}	1.0

proposed foreign synset(s):

- {vesel, voios, lipsit_de_griji, jovial, binedispus, fericit, entuziast, cu_veselie, cu_voioşie, cu_bucurie, cu_noroc, cu_fericire}
- {vesel, voios, lipsit_de_griji, jovial, binedispus, fericit, entuziast, cu_veselie, cu_voioşie, cu_bucurie, cu_jovialitate}

One notices that, in this case, the translation obtained when using the evaluation function defined in §2 is the most appropriate one.

Example no. 2

English synset: {intentionally, deliberately, designedly, on_purpose, purposely, advisedly, by_choice, by_design}

with intention ; in an intentional manner ; “he used that word intentionally” ; “I did this by choice”

e-sets:

eword	e-set	score
on_purpose	{dinadins}	7.0
purposely	{dinadins, intenţionat, cu_premeditare}	7.0
by_design	{cu_intenţie, intenţionat}	7.0
deliberately	{dinadins, intenţionat, voit, cu_premeditare}	6.0
designedly	{cu_intenţie, intenţionat, cu_bună_ştiinţă}	5.0
advisedly	{plănuit, intenţionat, cu_bună_ştiinţă}	4.666666666666666
intentionally	{intenţionat, voit, dinadins, anume, expres}	4.6000000000000005
advisedly	{după_matură_chibzuinţă}	1.0
advisedly	{judicious}	1.0
by_choice	{de_preferinţă}	1.0
by_choice	{cu_precădere, îndeosebi}	1.0
deliberately	{gândit, cu_grijă, atent}	0.6666666666666666
deliberately	{încet, fără_grabă}	0.5

proposed foreign synset(s):

- {plănuit, intenţionat, cu_bună_ştiinţă, de_preferinţă, cu_intenţie, dinadins, voit, cu_premeditare, anume, expres}
- {plănuit, intenţionat, cu_bună_ştiinţă, cu_precădere, îndeosebi, cu_intenţie, dinadins, voit, cu_premeditare, anume, expres}

English synset: {intentionally, deliberately, designedly, on_purpose, purposely, advisedly, by_choice, by_design}

with intention ; in an intentional manner ; “he used that word intentionally” ; “I did this by choice”

e-sets:

eword	e-set	score
deliberately	{dinadins, intenţionat, voit, cu_premeditare}	24.0
intentionally	{intenţionat, voit, dinadins, anume, expres}	23.0
purposely	{dinadins, intenţionat, cu_premeditare}	21.0
designedly	{cu_intenţie, intenţionat, cu_bună_ştiinţă}	15.0

advisedly	{plănuit, intenționat, cu_bună_știință}	14.0
by_design	{cu_intenție, intenționat}	14.0
on_purpose	{dinadins}	7.0
deliberately	{gândit, cu_grijă, atent}	2.0
by_choice	{cu_precădere, îndeosebi}	2.0
deliberately	{încet, fară_grabă}	1.0
advisedly	{judicious}	1.0
advisedly	{după_matură_chibzuință}	1.0
by_choice	{de_preferință}	1.0

proposed foreign synset(s):

- {plănuit, intenționat, cu_bună_știință, cu_precădere, îndeosebi, cu_intenție, dinadins, voit, cu_premeditare, anume, expres}

However, in the case of this English synset, the translation obtained when using the other evaluation function seems more appropriate.

None of the generated translations are entirely correct, which leads to the assumption that better results could be obtained when using a completely different evaluation function.

This type of result reinforces once again the necessity of combining class methods with structural ones, whenever possible, when performing this type of task, while pointing out the importance of the evaluation function used by Algorithm 2.1 and its variants. It becomes obvious that other evaluation functions should be conceived and tested as well, within future studies.

9. FINAL REMARKS

The proposed approach to WN generation is a combination of automatic and manual methods. The manual method relies on human experts, while the automatic one strongly relies on bilingual dictionaries and represents a combination of class methods with structural ones.

The significance of the manual effort involved in quality assurance primarily depends on the existence of appropriate tools. The involved human effort can be greatly reduced in the case of those languages for which correct and complete bilingual dictionaries in electronic format exist. Thus, in most cases when the obtained results are not the best possible ones, it is mainly because of the imperfection of existing bilingual dictionaries.

The main difficulties which occurred when automatically translating English synsets into Romanian ones were generated by collocations, by loan translation, and by the fact that the polysemy of many English words is greatly superior to that of the corresponding Romanian words. No matter what language is taken into account, linguistic difficulties that can not be overcome will always exist. Additionally, we should note that most problems occur when translating English synsets that contain a single polysemous word, Algorithm 2.1 being unable to decide among meanings. Such synsets were subject to further investigation and the enrichment step taking into account the WN structure was added to Algorithm 2.1 whenever possible. Our improved method reinforces the necessity of combining class methods with structural ones when dealing with this type of task. It has also become obvious that other evaluation functions should be investigated in future studies.

Finally, we would like to emphasize the fact that, in the absence of truly competitive tools (with reference to both paper and electronic dictionaries) the realistic evaluation of the proposed algorithms and of the corresponding computer programs becomes rather difficult, if not almost impossible.

Acknowledgements. The present research was initiated within the framework of the BALRIC-LING project, funded by the European Commission as part of the IST Program (IST-2000-26454), and was concluded during the Fulbright Grant running February 1 – August 1, 2004 at the Cognitive Science Laboratory of Princeton University. The author would like to thank both the European Commission and the Romanian - U.S. Fulbright Commission for the importance they have attached to the presented topic, as well as for having offered their full support. For the purpose of updating this overview the author has performed the presented tests using the latest existing Prolog WN database package, namely that of WN 2.0.

REFERENCES

Atserias, J., S. Climent, X. Farreres, G. Rigau, H. Rodriguez, 1997, *Combining Multiple Methods for the Automatic Construction of Multi-lingual WordNets,* in: N. Nicolov, R. Mitkov (eds), *Recent Advances in Natural Language Processing II. Selected papers from RANLP '97,* John Benjamins Publishing Company, Amsterdam/Philadelphia, 327–338.

Fellbaum, C. (ed.), 1998, *WordNet: An Electronic Lexical Database*, Cambridge/London, The MIT Press.

Harabagiu, S., 1999, *Lexical Acquisition for a Romanian WordNet*, in *Proceedings. EUROLAN '99*, Iaşi.

Hristea, Th., 2003, *Some Linguistic Comments Concerning the Obtained Output*, in F. Hristea, M. Popescu (eds) *Building Awareness in Language Technology*, Bucharest, Editura Universităţii din Bucureşti, 153–157.

Hristea, F., 2002, *On the Semiautomatic Generation of WordNet Type Synsets and Clusters*, *Journal of Universal Computer Science*, 8, 12, 1047–1064.

Hristea, F., 2003, *On the Semiautomatic Generation of Verb Synsets in Languages other than English*, in *Annals of the University of Bucharest*, LII, 75–86.

Hristea, F., C. Vaţă, 2005, *An Algorithm for the Semiautomatic Generation of WordNet Type Synsets with Special Reference to Romanian*, in Proceedings of the Workshop "Language and Speech Infrastructure for Information Access in the Balkan Countries", in conjunction with RANLP – 2005 (Recent Advances in Natural Language Processing) Borovets, Bulgaria, 23–30.

Levitchi, L., 1973, *Dicţionar român-englez* (3-rd edition), Bucharest, Editura Ştiinţifică.

Levitchi, L., A. Bantaş, A. Nicolescu, 1974, *Dicţionar englez-român*, Bucharest, Editura Academiei Române.

Miller, G. A., R. Beckwith, C. Fellbaum, D. Gross, K. J. Miller, 1990, *Introduction to WordNet: an on-line lexical database*, *International Journal of Lexicography*, 3, 4, 235–244.

Nikolov, T., K. Petrova, 2000, *Building and Evaluating a Core of Bulgarian WordNet for Nouns*, *OntoLex '2000 Report*, Sozopol, Bulgaria.

Nikolov, T., K. Petrova, 2001, *Towards Building Bulgarian WordNet*, in *Proceedings of RANLP '01*, INCOMA Ltd., Tzigov Chark, Bulgaria, 199–203.

Sima, C., C. Vaţă, 2004, *On the Semiautomatic Generation of Romanian Noun Synsets*, in *Annals of the University of Bucharest*, LIII, 1, 125–136.