The ZStation as a computational aid for terminography

Henri Zinglé,
Laboratoire d'Ingènierie Linguistique,
Universitè de Nice, Sophia Antipolis (F).

1. About the ZStation

The ZStation (Zinglé, 1994) was initially developed as a workbench to help linguists without real experience in programmation to develop linguistic resources for advanced natural language processing and applications based on it. This tool is characterized by:

- its user-friendliness (Windows interface)
- its high interactivity (modelisation can be tested immediately)
- the reusability of the developed resources
- the inherent multilingual approach (due to the priority given to semantic description).

In this system linguistic data are organized into to levels: a) the *interlinguistic level,* which is based on a description of concepts and relations between concepts (either encyclopedic or specific ie bound to domain knowledge) b) t*he intralinguistic level* which takes into account the linguistic knoowledge related to a particular language (inflexion, derivation, composition, syntax etc...).

In addition to classical functions of file managing and editing of common computer applications the ZStation provides 3 major managers related respectively to resources, applications and tools. Figure 1 in appendix shows the general menu of the ZStation.

The *resource manager* allows users to create and to test ontologies, dictionaries, inflexional grammars, syntagmatic grammars and specification micrograms for derivation, composition, homograph resolution, extraction of structures etc...

The *application manager* provides for the moment lexical analysis and generation, textual analysis, analysis of sentences into conceptual graphs, automatic generation of multilingual glossaries, document indexation as well as statistics and concordances. The list of applications is not closed and new applications may be added in the future.

The *tool manager* offers a large number of facilities to assist linguists in the development of linguistic resources. In particular they may list and count forms or lemmas from corpora, extract automatically compounds and phraseologic units from corpora, build or complete domain dictionaries from corpora. Likely to the precedent manager the tool manager may include in the future additional functions in relation with the needs expressed by users.

The *ontologies* are based on a description of concepts using typed conceptual relations with an inheritance mechanism which takes into account the propagation of negation on the hyponyms. Relations are defined by users and are not strictly of generic type; they include also modality (necessary vs facultative relations). The elaboration of ontologies is not a very easy task, but it is fundamental as it determines the links between languages (concepts are universal even if all concepts are not actualized in all languages). Users fix the level of description as

the granularity of knowledge has to be choosen close to the domains of application and not in the absolute –to what linguists natural tend.

The *dictionaries* are based on a description of lemmas in regard to their semantic, inflexional and syntactic properties. A lemma may be connected to different descriptions. The french form ferme may be an adjective *(la poire est ferme)*, a noun *(Jean travaille ‡ la ferme)* or a verb *(Jean ferme la porte)*; on the other hand the french noun glace may be linked to the concepts of "ice", "icecream", "mirror" or "car window>. Each description is composed at least of an concept identifier (from which the lemma inherits its semantic properties), a flexional identifier (from which the lemma inherits its morpho-syntactic properties) and a structural description. The structural description either consists in a list of syntactic-semantic properties or is limited to a syntactic-semantic identifier from which the lemma inherits its structural properties. The structural description allows to distinguish for example between "to steal" or "to fly" for the french verb voler *(Jean a volé une montre vs Les canaris volent)*. It is possible to generate a dictionary of forms from a dictionary of lemmas; this kind of dictionary which normally associate forms to the corresponding lemmas and to their morpho-syntactic variables are useful for analysis; the lemma is used as pointer to the semantic and syntactic-semantic properties.

The *inflexional grammars* define sets of morphosyntactic properties (category, gender, number, case, etc..) and link them to morpho-syntactic identifiers. The association of a lemma and of a morpho-syntactic identifier generates all the forms of the lemma which the grammar of the concerned language may produce. For example, the french lemma porte associated to the morpho-syntactic identifier femme (standing for all french nouns with feminine gender adding -s in the plural) will produces porte (category=noun, variables=feminine, singular) and portes (category=noun, variables=feminine, plural).

The *syntagmatic grammars* explicite the grammatical conditions which are bound to syntactic-semantic properties. They differ from classical NLP syntagmatic grammars, as the focus is on validation of the conditions which allow to link words together rather than on the calculus of constituents. As the system is basically object oriented, the syntagmatic grammar has to explicite the nature/function messages which are emitted by words during analysis. Let us suppose for example that the possible connection of the lemma glace to a word meaning a fruit is associated in its structural definition to a message {category=sub, function=modSub}; this message expresses the formal condition that the system has to look for a noun preceded by the preposition ‡ and modifying the noun glace; the sense of this message has to be explicited in the syntagmatic grammar using standard predicates related to basic linguistic operations such as precedence, congruence, access to morpho-syntactic information, etc.

The *specification micro-grammars* which are bound to tools or applications such as derivation, composition, homograph resolution and extraction of phraseology are mainly based on rewriting rules. These grammars define symbols using chains of symbols or links to morpho-syntactic knowledge. In the case of homo-

graph resolution the rules are context sensitive and allow to explore a variable context size around processed words.

Users have to respect a few conventions imposed by the system for data edition; these conventions vary in function of the resource type; insertion functions are provided, which reduce significantly errors during edition. Specific compilers check the correctness of the users' input and translate it into the data format adapted to the application processors. A test tool is provided for each resource which allows to verify immediately if the elaborated formalization produces the correct results. The tools for edition, compilation and test are language independent, so it is possible to proceed to the formalization of linguistic knowledge in various languages.

The system has been developed in PDC-Visual Prolog on the MS-WINDOWS platform.

2. Extracting data from corpora

2.1 Extracting lexical units

The tool manager of the ZStation provides three functions: extraction of simple units, extraction of compound forms and extraction of complex units using a specification grammar.

Figure 2 in appendix shows the dialog interface for the extraction of simple units. The selected corpus appears in the listbox; files can be added or removed using respectively the buttons "Ajouter" or "Supprimer". The result of the extraction will be put in the file specified in the input field "Fichier résultat"; a button associated to this field activates the normal file browser of WINDOWS. The extraction may be achieved using an existing dictionary of forms, which is selected in the same way as the result file. If a dictionary is selected for the extraction process, the words will be lemmatized. The words which are absent from the dictionary will be reproduced in the output as forms. In the result list all units are sorted with the indication of the number of occurrences for each unit. The result file may be edited by all kind of text processor on WINDOWS platform or with MS-WINDOWS applications such EXCEL or ACCESS.

The extraction of compound forms is based on the stochastic methods used for file compression (Bernstein, 1992). The extraction process looks for recurrent form sequences and build during text scanning a lookup dictionary in which each sequence is associated to its number of occurrences in the corpus. To each word or chain, which was found previously, is added the next word and the new sequence is inserted in the lookup dictionary for further matches. At the end of the process the content of the lookup dictionary is retrieved and solutions with a very low representation are eliminated. This method is very quick, as the text is processed in a single pass, but the solutions have to be revisited to eliminate recurrent sequences which are not compound forms. For the moment this process is work-

ing on the basis of forms and we hope to increase the quality of results by introducing the lemmatisation of forms; this modification will modifiy the number of occurrences of detected lexical units and contribute thus to have a better representation of significant sequences.

In a different way, the extraction of compounds uses linguistic knowledge to extract patterns from texts. This method is based on a specification grammar, within which users may describe the formal structures of compounds. Figure 3. in appendix shows an example of compound grammar using rewriting rules. This rules define symbols through attribute/value templates; the terminal rules use links to the morpho-syntactic properties of the processed language. Symbols preceded by underscore in the right part of the rule indicate that the words will be lemmatized in the output file. The algorithm is based on top-down phrase structure grammar using chart parsing technique; chart parsing is very useful here as it avoids to reexamine parts of rules which were recognized previously. The output file provides a sorted list of sequences associated to their occurrences in the corpus. Even this method is more attractive in a linguistic point of view, it is however by far slower than the extraction of compound forms. The result have also to be revisited, as the rules are sometimes insuffisant to distinguish between lexical units and phrases on a formal basis *(cf. Jean mange des pommes de terre vs Jean recouvre les pommes de terre).*

Users have to choose what method which is more adapted to their concrete work. The results may be incorporated to existing dictionaries or be used only for lexicological investigation. Presently we are developing methods for term extraction comparing the result of the extraction of lexical units with information stored in dictionaries of standard language.

2.2 Extracting phraseology

The extraction of phraseologic units offers similarities to the extraction of compounds. Two methods were developed using respectively pure stochastic methods and specification grammars.

The stochastic method extract in a first stage all sentences of the corpus and put them in a database; in a second stage the extraction algorithm builds candidate sequences by comparing all intersections of sentences in the corpus. The method does not look only for contiguous elements but for all elements which appears in a given order in two or more sentences. Filtering methods are under development, which should reduce the number of solutions produced by the algorithm.

The second method is similar to that used for compound detection. The user has first to develop a specification grammar. The rules are of a phraseologic grammar are similar to these of a compound grammar; however in a phraseologic grammar the elements are not necessarily contiguous. The grammar is based on rewriting rules and the algorithm uses also top-down chart parsing.

2.3 Text analysis

In addition to both methods presented above the ZStation offers the possibility to perform textual analysis ie to look for specified information in texts. This process is activated through the application manager. A document retrieval language was developed using basic commands which users may combine to scan texts. About ten basic commands are available. To look, for example, for a noun indicating an animal combined to a verb indicating an aerial travel mode the sequence of commands will be following:

{match([N,V]&cat(N,su)&cpt(N,isa,+animal0)&cat(V,vb)&cpt(V,isa,+fly0)&accord(sv,N,V)}

This method may be used for the detection of phraseologic units. It is however more complicated than pure phraseology detection with a specification grammar, as it needs a lot of linguistic resources. Figure 4 shows the interface for text analysis.

3. Building indices and concordances

An other way to extract data for corpus investigation is to build indices or corpora. The application manager provides here tree types of function: building of indices, querying of indices, building of stactistics and concordances.

3.1 Building indices

Users have the choice between following options:

- *word indices*: this process scans the corpus for all references found for each lexical unit in the text; a dictionary can be used for lemmatization; lexical units which are not found in the dictionary are inserted as forms in the index;
- *lemma indices*: this process looks for all references on one or several lemma(s) typed in by the user; it uses also a dictionary of forms to lemmatize words found in the text; the references point to all forms in the corpus which are related to the specified lemma(s);
- *category indices*: this process calculates all references to one or several morpho-syntactic category (ies), for example to all nouns or all adjectives in the corpus; this process uses also a dictionary of forms;
- *concept indices*: this process scans the corpus for all words related to one or several concept(s); it is thus possible for example to look for lexical units expressing colours, animals, vegetables, violent actions, etc.;
- *formant indices*: this process looks for all lexical units containing one or several formant (prefix, suffix, infix or stem); it is generally useful for lexicologic investigation.

An individual dialog form is associated to each procedure and users have to select the parameters and the linguistic resources required in each case. After validation of the form the corresponding procedure is automatically executed. Users specify the content of the corpus using the standard procedure of text selection. The processed file names are registered to avoid that files are processed twice (what have an incidence of statistics).

3.2 Querying indices

The ZStation provides a function with allows to consult indices. The query interface allows users to type in the information they are looking for and shows the result of the database lookup in an edit field. The result consist in the indication of the exact source document and the position within it. As querying indices is interactive, it has normally be used to control the result of indexation. To list the content of indices the procedures indicated in ß 3.3 are recommended.

3.3 Building of statistics and concordances

Indices are batabases programmed in Visual Prolog in which data are encoded. The contend of indices are therefore not directly readable and have to be converted for lexicologic or lexicographic use.[1] Users may chose to convert the content of an given index either into a statistical repository or into a concordance.

In the first case all information items are listed in alphabetical order with following indications for each item (cf. figure 6 in appendix):

- the sum of all references in the corpus;
- the sum of all references in each textfile of the corpus;
- the list of all references in each textfile.

The statistical repository may be edited by any word processor running on the Windows platform or the data downloaded to Excel or Access.

In a similar way indices may be converted into concordances. Information items are also sorted in alphabetical order with following indications for each item:

- the file identifier;
- the sentences in which the item occured.

The result file may be processed by usual tools. In a lexicologic point of view concordances are very useful, as they allow users for example to access directly the context of lexical units.

1-It may be however directly used for document retrieval.

4. Building dictionaries and multilingual glossaries

4.1 Creation of new dictionaries

The tool manager of the ZStation provides the possibility to create domain dictionaries which may be used later for terminologic work or for document retrieval. Users can spare a lot of time using this facility, as all the knowledge provided by an existing dictionary can be reused. The interface (cf. figure 8 in appendix) allows to select the corpus, which may be composed of one or several textfiles, to choose the name of the dictionary to be created, to indicate the existing reference dictionaries (ie a dictionary of forms and a lemma dictionary of basic language) which should be used. Words which already exist in the reference dictionaries are lemmatized and all intralinguistic information about them is automatically inserted in the new dictionary. The forms which are not recognized are inserted in the new dictionary and marked up with the symbol <*>. After the whole corpus has been scanned, users may use the normal dictionary edit functions to modify the dictionary entries related to unrecognized forms. If an unrecognized dictionary entry corresponds to the lemma, only the symbol <*> has to be suppressed otherwise the dictionary entry has to be renamed; in both cases all needed linguistic information has to be typed in.

4.2 Incrementation of dictionaries

If the corpus chosen for the creation of a new dictionary reveals to be not significant enough in practice, users may modify the content of the dictionary by processing additional corpus. The method is quite similar to the creation of new dictionaries with this difference that in this case the reference dictionaries are not dictiona-ries of standard language but previously created domain dictionaries. The forms which cannot be recognized on the basis of the referent dictionaries are also inserted with the symbol '*' and the dictionary entries have to be revisited manually in a later stage.

4.3 Creation of multilingual glossaries from monolingual dictionaries

The creation of multilingual glossaries which is a further tool provided by the ZStation is slightly different from both preceding procedures as it is not based on corpus examination. The method here intends to parallelize lexical units belonging to different languages using the semantic identifiers of dictionary entries. For example, the spanish lemma manzana and the french lemma *pomme* may be considered as equivalent as both refer to the concept "apple".[2]

A mutlilingual glossary is build by scanning two or more dictionaries for all entries which have the same concept identifier. The first dictionary in the the list of

2- In the notation used in the ZStation apple0 represents the concept of apple and not the lemma apple. Concept identifiers are marked up by a number in opposition to lemma identifiers which appear like in usual dictionaries. Even if users can choose concept identifiers in their own language, it is preferable - in our point of view- to choose identifiers in english to insure the readability of dictionary codification for all users.

selected dictionaries is assumed to furnish the glossary entries. This is here a very convenient way to build multilingual glossaries, but users have to keep in mind that the result is depending from the quality of semantic codification. Multilingual gloassaries only list lexical units for the moment, but we hope to introduce additional informations about formal properties of lexical units as well as definitions derived from the ontologic representation of underlying concepts.

5. Analyzing and generating neologisms

The survey of neologisms occurring in corpora is important for terminology. If they are well formed and viable in linguistic point of view, they may be introduced later in official repositories or proposed for computer aided translation. As specialists of terminology are not experts of all domains they encounter, it seems thus useful to propose a tool which is able to give a rough interpretation of the neologisms on a pure lexicological basis including formal as well as semantic features. This interpretation may motivate an eventual validation by domain experts. On the other hand, the generation of neologisms appears to be useful in terminologic planing. We propose here a tool for the generation of neologisms which builds terms from a semantic description.

Both applications are based on a standard knowledge base about formants, in which formal and semantic features concerning derivation (and composition) are explicited. To each format in of the base are associated a category identifier (prefix, sufix, infix, stem), a semantic description, the indication of the morpho-syntactic category the format is able to produce and a set of formal and semantic constraints to be satisfied. The semantic representation uses a functional formalism which can be eventually translated in conceptual graphs (Zinglé, 1996). The satisfaction of the semantic constraints uses information extracted from ontologies.

The lexical analysis (cf. figure 10 in appendix) calculates for each term its morpho-syntactic category and its meaning expressed in the above mentioned formalism. For example, the french word *enregistreur* will be analysed as follows

sub: agent (record0)	"a person which is recording"
sub: device0 (record0)	"a device used for records"

In the future we will also generate definitions in natural language from the semantic representation.

Inversely the generation produces words from a semantic representation typed in by users and the indication of the morpho-syntactic category of the word(s) to be created. Figure 11 in appendix shows an example of lexical generation. The notation *possible0 (record0)* produces her the derivate *enregistrable* for the category adjective and *enregistrabilité* for the catgeory noun. It is obvious that the method allows the generation of terms in different languages from a single semantic representation if the respective knowledge bases are available.

Concluding remarks

Initially designed for pure language processing using techniques of artificial intelligence the ZStation reveals to be now a very useful tool for lexicologic and lexicographic investigation. In difference to existing software for corpus analysis, the ZStation offers large possibilities for corpus analysis and data extraction far beyond usual form oriented text processing and statistical investigation. This is mainly due to the focus on linguistic resources. The ZStation is currently tested by several user groups and we are waiting now for feedback and suggestions, which will be important for the future evolution of the software.

References

BERNSTEIN D. J., (1992) Y coding.

SOWA J., *Conceptual structures: information processing in man and machine,* Addison-Wesley, 1984.

ZINGLÉ H., *The ZStation workbench and the modelling of linguistic knowledge. Current Issues in Mathematical Linguistics,* Elsevier-NHLS, 1994, pp. 423-432.

ZINGLÉ H., *Analyse et génération automatiques de mots dérivés en francais,* Travaux du Lilla, n° 81, 1995.

ZINGLÉ H., ZART: *Un linguiciel d'aide la rédaction scientifique et technique en langue étrangëre,* Travaux du Lilla, n° 81, 1995.