Reusability as easy Adaptability: a substantial advance in NL Technology

Irina Prodanof, Amedeo Cappelli, Lorenzo Moretti

Istituto di Linguistica Computazionale - CNR
Via della Faggiola 32, 56100 Pisa
irina@ilc.pi.cnr.it

Abstract

The design and implementation of new applications in NLP at low costs mostly depends upon the availability of technologies oriented to the solution of any specific problem. The success of this task, besides the use of widely agreed formats and standards, relies upon at least two families of tools, those for managing and updating, and those for projecting an "application view-point" onto the data in the repository. This approach has different realizations if applied to a dictionary, a corpus, or a grammar. Some examples, taken frrom European and other industrial projects, show that reusability:

a) in the building of industrial prototypes consists in the easy reconfiguration of resources (dictionary and grammar), easy portability and easy recombination of tools, by means of simple APIs, as well as on different implementation platforms:

b) in the building of advanced applications still consists in the same features, together with the possibility of opening different view-points on dictionaries and grammars.

1. Introduction

The design and implementation of new applications in NLP at low costs mostly depends upon the availability of technologies oriented to the solution of any specific problem. The large interest into linguistic resources, like corpora, dictionaries, and collections of grammar rules is in general motivated by the assumption that natural language applications would become easy to implement if they can rely on the possibility of (i) inferring new linguistic forms from samples of real language and (ii) deriving the necessary (classificatory) information from some repository, dictionary or grammar, which contains all the possible information expressed in a generally agreed format.

Although the benefits of this approach have not been openly declared in any project, they are obvious, but success, besides the use of widely agreed formats and standards, relies upon further features which have not been fully stated. In fact, a plain resource shall be accompanied by at least two families of tools, those for managing and updating, and those for projecting an "application view-point" onto the data in the repository.

The idea that a resource can reach a "non-growth" point where ideally all the information needed for all the prospective applications is available, conflicts at least with the basic principle of good linguistics, non to speak of reality. Thus a linguistic resource is to be conceived of as a bunch of methodologies and procedures, where the repository of information and knowledge is only a part.

This approach has different realizations if applied to a dictionary, a corpus, or a grammar. For dictionaries it is necessary to be able to build updates from text corpora any time a new application is confronted. Thus, dictionary updating is a methodology rather than a single interface programme. Also, collections of grammar rules must be updatable by means of a reusable methodology, which can be reinstated at any new application.

The other function consists in the possibility of mapping different perspectives onto such repositories, according to both the requirements imposed by application domain and the view of the project staff. Dictionaries are to offer morphosyntactic information as input to POS tagging, as well as to syntactic parsing; but also different kinds of semantic information shall be retrieved for more in-depth analysis. Simple taxonomies for use in information retrieval, complex ontologies, and conceptual structures must also be extracted from a lexical repository. Grammars must be able to meet the requirements of different applications, but also of different theoretical approaches. For instance, sentence analysis can be carried at different levels of depth according to the type of application, while project staff may choose different types of representation according to the domain dealt with, the general architecture of the aimed system, or even the personal theoretical inclinations. Thus a grammatical resource should be expressed in a sort of metarepresentation, able to accommodate different grammatical formalisms; this is not intended to be another theoretical claim over some formalism, but simply a recommandation to establisha sort of low level format which does not impose constraints over grammatical formalisms.

2. Reusability: two dimensions

The existence of both comprehensive resources and robust and flexible tools does not imply, however, that they can be reused in any new application as they stand, just gluing pieces together. This requisite in nowadays language technology very much resembles the utopy of high quality fully automatic machine translation. Thus the notion of reusability shall not be taken for granted, but is problematic in itself.

Two dimensions of the problem will suggest the opportunity of searching for a realistic view of reusability. On one hand, a resource should be made sensitive to variations through time, domain, and social setting. On the other hand, a tool should be made responsive to demands arising from applications not previously foreseen and classified (in a view of industrial expansion). The answer to the first point consists in the development of efficient and robust methodologies for modifications (including total change of bodies of knowledge). By methodology a collection of strictly modular tools, together with the possibility of chaining them in accordance with any specific application is meant.

The second problem is the one of customization, and can be solved by devicing application-oriented recombinations of tools and resources: this calls again for modularity and for application-driven heuristics of use.

3. In-depth cases

Many projects have been devoted to the specification of the means to allow an easy reuse of linguistic resources. In particular, TIPSTER (Grishman, 1996) and GATE (Peters et al., 1998; Cunningham et al., 1998) 0realized basic software tools for the integration of different non omogeneous linguistic resources, by the use of a core object oriented model, yelding a set of resource-specific inheritence hierarchies. These projects heavily exploit the standardization of linguistic concepts.

The following examples from some European and other industrial projects show that reusability:

a) in the building of industrial prototypes consists in the easy reconfiguration of resources (dictionary and grammar), easy portability and easy recombination of tools, by means of simple APIs, as well as on different implementation platforms:

b) in the building of advanced applications still consists in the same features, together with the possibility of opening different view-points on dictionaries and grammars.

3.1. CRISTAL: industrial prototype

LRE 62-059 CRISTAL, finished in 1998,was a project on multilingual access to textual documents, intended to retrieve information according to a search-by-an-idea approach. Text were classified by a linguistic analysis procedure and queries could be expressed in natural language.

MIRA, the natural language front-end developed by the Italian team, has been developed as an independent component to be plugged into CRISTAL, and consists of a Dictionary, a Grammar, a Preprocessor and a Postprocessor.

Its core engine, NLGRADE, is a bottom-up all-paths algorithm, designed to run APSG formalisms, i.e. ordinary reduction rules enriched with augmentations, which may include feature structure management and semantic actions. The output is a tree or a set of trees, if the sentence is ambiguous, expressed in terms of a parse graph structure. MIRA integrates different software platforms, often imposed by efficiency requirements, but is designed to work in a C environment. The Dictionary is a domain-oriented subset of the DMI and is too large (31,530) to be accessed directly by the parser; so, it is managed by a Relational DBMS. The Preprocessor, written in ANSI C, is in charge of activating such a DBMS and converting the input strings into lexically interpreted sentences, expressed in terms of list-structures, compatible with the core parser. The output of the parser is also in terms of lists, be it a single or a set of trees, but the Postprocessor selects the most plausible tree and converts it into C++ objects to be mapped onto the conceptual structures of the Dicologique conceptual dictionary (Dutoit, 1992). It also uses heuristics to give only one tree in output and, if the sentence is ungrammatical, highlight the partial analyses with maximal coverage

MIRA configures the integration of different general purpose modues, both home-made, like NLGRADE and DMI, and commercial, like Fulcrum. All modules are specialized to the specific application, both from the point of view of the repository of data, and in terms of software. This realizes a sort of "basic level" reusability.

3.2. Getting more advanced: ACQUILEX

The system PALCO has been developed, out of the same components of the previously described CRISTAL, in the frame of the LRE project ACQUILEX II. The general objective of ACQUILEX is the acquisition of lexical knowledge from non annotated corpora; thus, the central task is the analysis of unrestricted texts. To this purpose, no selection has been imposed onto the dictionary, which is the entire DMI in its complete extension, managed by the same DBMS, activated by a slightly modified Preprocessor. This has been extended to treat text-specific features like punctuation, special characters, figures etc.

Dealing with unrestricted texts, the major problem of the parser is the huge number of syntactic trees associated to each sentence. Given the properties of the core parser, this does not cause any processing inefficiency, but the real problem is the readability of the resulting structures. Thus the task of the Postprocessor has been extended to packaging the output trees and applying a top-down filtering, in order to produce a minimal analysis set for each sentence.

PALCO reuses the same dictionary, with no domain dependent restrictions, and the same core parser, while Preprocessor and Postprocessor have been slightly modified. The great gain of PALCO is the implementation of a complete grammar for unrestricted text, with wide coverage.

3.3. A conceptual dictionary for TAMIC-P

LE TAMIC-P aims at the creation of a system which allows a transparent and efficient access to multiple databases in the domain of Public Administration. It also admits queries expressed in Italian, but, in this case, a different parser has been used (Bagnasco et al., forthcoming). The dictionary, instead, is the previously described DMI, a subset of 21775 items has been extracted from a corpus of 6.3 Mb of circular letters of the National Social Security Agency (INPS), and morphosyntactic information has been automatically translated into the codes and formats compatible with the chosen parser.

In addition, TAMIC requires the use of a conceptual dictionary, in the style of WordNet; thus, also the storing format of the DMI has been modified in such a way as to allow a WordNet type treatment, including retrieval of a word meaning, of synonyms, hyperonyms, hyponyms, and meronyms. Nevertheless, the basic data-base management stick the same relational model.

The interface to the whole system is realized with hypertext technology. It allows interaction with the corpus as direct access, indexing of the corpus and viewing of the corpus from a single dictionary item.

Search can be carried by combining pairs of words by using boolean operators (AND, OR, NOT).

A hypertextual technology is used also for the creation and management of WordNet relations among words in the dictionary. Different "access forms" allow a guided access to the data-base, in order to carry the basic operations of:

- creating a new synset, together with insertion of synonyms and hyperonyms, and equivalents in German and English;
- introducing subcategorizations and logical forms;
- connecting further elements in a hierarchy.

From a technical viewpoint, the system is organized into two different environments: the Data Base in which data concerning the conceptual dictionary are stored and the hypertextual interface in which the forms have been realized. The access is enabled by using a language containing instructions both of the SQL language and of the interface environment. In this way we can benefit from the flexibility and hypertextuality of the interface and the efficiency of the DB. The hypertextual environment controls the queries and amplifies the use of the SQL (Cappelli & Moretti, 1999). The system has been implemented in Macintosh Apple; the interface in Hypercard, and the Data Base in BUTLER SQL 2.5.2..

The result of the construction of the technical dictionary has gone beyond the initial goals of TAMIC, since it is not only a linguistic resource to be used by the parser but it is also an autonomous system to be used for different purposes. It is the focal point of TAMIC since it makes it possible to connect a rich variety of linguistic expressions to the conceptual representation of the domain of social security, in this way allowing the interpretation of a query on the DBs which store individual data of the domain itself. The data of the dictionary can also account for many aspects of the INPS activity and they do not confine themselves to the strict domain of pensions.

The system has some characteristics of usability which make it suitable, with minor modifications, for further applications such as:

- the translation, both automatic and human, in which a structured terminological multilingual dictionary can be used as a module in an automatic system or as an help for an human translator by giving the possibility of rationally retrieving several lexical options which refer to a concept;
- the textual information retrieval for amplifying the range of linguistic terms used in a query or for refining the meanings of technical terms for a large non expert public;
- the documentation in general, for storing and retrieving in a systematic way any material pertinent to an administration, also by using multilingual modalities which are very important in the contemporary global organization of affairs;
- the creation of standard texts of every type, as an help in the choice of the most appropriate lexical items.

3.4. Refining reusability: SPARKLE

The project LE SPARKLE is a sort of further refinement of reusability. Its objective is the creation of a sort of methodology to acquire syntactic information from annotated texts in order to improve expressivity and performance of a syntactic analyzer. Thus, like in ACQUILEX, the first task is to parse unrestricted texts, but the consequences should result in a feed-back of information to the parser itself.

Thus, the general schema is exactly the same as in ACQUILEX: the complete DMI of 100,000 entries is used, the Preprocessor treats all the specific textual phenomena, and the Postprocessor reduces parse forests to minimal analyses. The grammar is also the same, with the same coverage as in PALCO, with the exception that it has converted into a form which follows EAGLE recommendations.

The aim of the SPARKLE project was to study in which ways a parsing system can be improved by "lexicalization", i.e. providing more lexical information to the system. To reach this goal, some formal methods of evaluation were defined, in order to precisely calculate the precision and the recall of the analyzers at their different stages of development and lexicalization (Carrol et al., 1997).

The experiment carried on was slightly different with respect to the activities of the other partners in the project; the normal schema of development foreseen was to collect lexical data about the subcategorization of verbs starting from tagged corpora, and to apply these data to stochastic parsers.

Actually, our analyzer is a rule-based parser, so lexical data had to be translated into phrase structures and augmentations for the grammar rules; furthermore, lexical data were automatically extracted from a corpus by means of a "chunker", obtaining a lexical base of subcategorized verbs which is looked up by the parser at run-time.

In the framework of this project, two different level of analysis were studied: the phrasal level and the grammatical relation level; the two levels are shown in the following figure: the syntactic tree represent the first level of analysis and the feature structure of the root node represents the second one:

```
Sentence: il mercato chiede nuove regole
Running PG: Palco4
Total Parsing Time: 0.016 secs. (16 ticks)

 (S[15,0]/S2-1 = "IL MERCATO CHIEDE NUOVE REGOLE"
        (Np[7,0]/Np17 = "IL MERCATO"
                (Artdef[1,0]/Dictionary = "IL")
                (N[2,2]/Dictionary = "MERCATO"))
        (Vg[8,0]/Vg1    =    "CHIEDE"    (V[3,0]/Dictionary    =
"CHIEDE"))
        (Np[14,0]/Np14 = "NUOVE REGOLE"
                (Adjp[10,0]/Adjp2 = "NUOVE"
                        (Adjqual[4,1]/Dictionary            =
"NUOVE"))
                (N[5,0]/Dictionary = "REGOLE")))

Features of node [15,0]:
[ Subj ] : [ "CHIEDERE" "MERCATO" ]
[ Obj ] : [ "CHIEDERE" "REGOLA" ]
```

The phrasal level was considered a preliminary one; the final evaluation was due for the grammatical relations level only.

The parser needed no major modifications in order to use the data coming from the lexical acquisition process. Since each rule in the grammar is applied as an independent process, we had the possibility to constrain each rule application by means of a preliminary test. This test deals with the matching between the reduction set of the current rule and information made available through the lexical acquisition process.

For each verbal head in the test corpus (about 400 units) the lexical data were automatically extracted (about 5800 units), starting from the chunked form of the sentences. For each verbal lemma, some basic features (the possibility to support transitive use, predicative arguments, passive form) are given, along with a list of subcats.

Here follows a sample frame:

```
(lemma ABITUARE
        (PoS V)
        (trans YES)
        (pred NO)
        (pass NO)
        (subcat-list (
        ( (HEAD "encl" nil) (P_C IObj "A"))
        ( (HEAD nil "ESSERE") (I_C XComp "A")))))
```

A subcat is thus constituted by a verbal head and a list of argumental positions; the verbal head is represented through a triple formed by:

the place holder ("HEAD");
the auxiliary verb (if any) used in the particular syntactic context the subcat is extracted from;
the indication of (possible) clitic units;

each argumental position is a triple formed by:

the chunk type;
the grammatical function;
the selected preposition (for prepositional chunks only).

The lexicalization of the parser has taken two main steps: the updating of the grammar with the structure derived from the lexical acquisition process, and the linking between the parser processor and the acquired data.

- First step of lexicalization and evaluation baselines

Starting from the collection of the acquired subcats, we have extracted about 80 different structures for the sentence level which were missing in the grammar. These structures have been obtained through the definition of some simple principles of translation from chunks into phrases, i.e. from subcats into phrasal rules:

This work had the direct effect of widening the grammar coverage.

At this stage, based on the manual annotation of the test corpora, it was possible to give the baseline evaluation of our system.

The baseline for the phrasal level is hown in the following:

Phrasal level - baseline:
Sentences: 200
Nodes in the annotated corpus: 5904
Nodes returned by the parser: 2666
Total number of matching: 1236
Recall: 0.2
Precision: 0.5

In order to evaluate the behaviour of the system in the task of recognizing the grammatical relations among sentence constituents, grammatical relations were assigned by default, depending on the syntactic type of the constituent and using the underspecified relation ``DARG" for both the "subject" and "direct object" functions. The resulting values are given in the following table:

Gramm. Relations level - baseline:
Sentences: 119
Nodes in the annotated corpus: 331
Nodes returned by the parser: 510
Total number of matching: 180
Recall: 0,5
Precision:0,4

- Second step of lexicalization and final evaluation

The second step of the lexicalization, which can be considered the most significant, was the implementation of the matching functions which control the construction of the sentence level nodes.

In a first experiment, the lexical data were used as constraints over the application of the grammar rules: each rule were applied if its reduction set had an exact match with at least one of the subcats extracted for the main verb of the sentence. After this modification, we observed a dramatic decrease in the coverage of the analyzer (which was 60% before, and 39% after lexicalization). This had an obvious theoretical reason: an "axiomatic" grammar is by definition designed for wide coverage, the more if it has been extended with those sentence patterns which had been extracted in the first step of lexicalization. But, adding the "subcat filter", a step very similar to the insertion of a functional control to a phrase structure grammar, results necessarily in a narrowing of the coverage, unless it is possible to associate at least one subcat to each production. This was not the case, as in the lexical acquisition phase statistical motivations suggested several criteria to restrict the number of subcats to acquire for each verb; for instance, low frequency or (so-called) noisy or discontinuous patterns were rejected. Evaluation at this stage was less significant, since we had no reliable method to analyze non parsed sentences, which constituted the greater part of the test corpus; taking into account parsed sentences only, we obtained the following results:

Phrasal level - mid:

Sentences: 98
Nodes in the annotated corpus: 2502
Nodes returned by the parser: 2680
Total number of matching: 1576
Recall: 0,6
Precision: 0,6

Gramm. Relations level - mid:

Sentences: 119
Nodes in the annotated corpus: 331
Nodes returned by the parser: 410
Total number of matching: 201
Recall: 0,6
Precision: 0,5

Anyway, the experiment could be nonetheless interesting: trying to give the final evaluation for the phrasal level analyzer, and assuming that our major problem was given by the decrease of grammatical coverage, we compared the evaluation results of the two systems taking into account evaluating only the actually parsed corpus. As shown in the table below, we could observe a general improvement.

Evaluation of parsed sentences only at the phrasal level

before lexicalization (baseline)	after lexicalization (mid)
recall 0,4	recall 0,6
precision 0,5	precision 0,6

This experience leaded us to the final implementation, which applies the following schema:

All the acquired subcats are stored in a data structure, using the verbal lemma as an entry key. For each grammar rule which recognizes sentence structures, a test function was added. This function recovers the verbal head looking at the feature structure of the main verb group of the current reduction set and compares the whole reduction set to the subcats of the corresponding verbal lemma; the application of the rule is performed if the reduction set occurs as a substring of any subcat; otherwise the rule is not applied or some recovery actions are performed, accordingly to the user's choice. When the rule is applied, functional information assigned to the chunks in the verbal subcat is retrieved and translated into a feature structure; this feature structure represent the recognized grammatical relations instantiated by the current node.

In this way we could effectively observe how lexicalization can improve the parsing system, and the resulting data do not suffer from the problems of a still in progress acquisition or of specific criteria adopted in it (such as, the pruning of undesired subcats).

Gramm. Relations level - final:

Sentences: 99
Nodes in the annotated corpus: 231
Nodes returned by the parser: 240
Total number of matching: 189
Recall: 0,8
Precision: 0,8

Evaluation at the Grammatical Relations level - synopsis

baseline	mid-term	final

recall 0,5	recall 0,6	recall 0,8
precision 0,4	precision 0,5	precision 0,8

In the final development of the system, we used a special flag to tag the recognized structures according to the kind of match between the structure itself and the corresponding subcats in the lexical database. A totally informal, but interesting datum, is the high level of linguistic adequacy of the analyses which feature an exact lexical match, instead of a partial one. Evaluating only these cases, the recall value would be much higher (approximately, 0.9); however, only about 50% of sentences returns an exact match. Thus, other "lexicalization phases" should extend the system, to reach real robustness without getting a lower degree of precision; at the same time, a finer tuning between the parsing strategy and the principles of lexical acquisition should be investigated, in order to better control the loss of coverage of the system with respect to its non-lexicalized version.

The second good result of our experience is that we have demonstrated that the approach chosen (to implement the integration between a rule-based parser and a lexical data-base) has given a real improvement. Its main characteristic is the high level of modularity and customisability: in fact, the system allows the user to trig on or off the various modules, tailoring its behaviour. Furthermore, this schema of architecture allows to integrate other modules for the treatment of other sets of information, such as semantic data.

The cases of wrong bracketing or attachment have been picked up by a single reviewer, basing upon his own sensibility and judgment, and taking into a major account the general specifications of the project. Proliferation of analyses is mostly due to all those cases in which a homograph term is not embedded into a major phrase; we have thus, in standard cases, a factor of multiplication which is a function of the number of such terms occurring in the sentence and the number of readings each of them has. We have already experimented in the ACQUILEX II project that these cases will have a much lesser rate of occurrence when the grammar is extended with the recognition of sentential structures.

Partial matching is considered a valid condition, in order to avoid a too strict constraint for the treatment of non necessary arguments of the main verb. The phenomenon of arguments not phonologically realized is in fact quite common in Italian, especially for the subject position. This may cause the extraction of lexical information from the corpus to return possibly incomplete subcategorization structures.

The final evaluation was due for the grammatical relation level only.

These new phases could address two different problems: to carry on the acquisition from larger corpora, to get new data, and to take into consideration other parts of speech (particularly, nouns and adjectives).

4. Conclusions

It turns out, thus, that:

- reusables are such, provided that they are built in a modular way and they can be constantly recombined to form always new applications;
- reusability is, in fact, only a matter of easy adaptability to new domains and new problems;
- reusability arises from a delicate balance between stable and large data repositories and procedures to use them;
- reusability is not a static concept, but the design of more pointed algorithms, and strong methodologies can make reusables more and more reusable with the time going.

Bibliographical References

Bagnasco C., Cappelli A., Magnini B., Zamatteo D., Accesso in linguaggio naturale ai dati della Pubblica Amministrazione: il sistema TAMIC-P, in *Atti del Sesto Congresso dell'Associazione Italiana per l'Intelligenza Artificiale*, (forthcoming).

Cappelli A. , Moretti L., TAMIC-P. Italian Technical Dictionary; Tamic-P Deliverable D6.0.4, Pisa, 1998.

Cappelli A., Prodanof I., Overview of the Natural Language Front-end of CRISTAL, *Atti del 5° Convegno "Cibernetica e Machine Learning" della Associazione Italiana per l'Intelligenza Artificiale*, Napoli, 1996.

Carroll J., Briscoe T., Carroll G., Light M., Prescher D., Rooth M., Federici S., Montemagni S., Pirrelli V., Prodanof I.,. Vannocchi M, Sparkle Work Package 3: Phrasal Parsing Software. Deliverable D3.2, 1997

Cunningham H., Humphreys K., Gaizauskas R., Stower M., CREOLE Developer's Manual, Univesity of Sheffield, 1998

Dutoit D., A Set-Theoretic Approach to Lexical Semantics, *Proceedings of COLING-92*, 1992, 982-987

Ferrari, G., Mac Aogain, E., Marino, M., Prodanof, I., Reilly, R., Saffiotti, A., Sheehy, N., CFID: A Robust Man-Machine Interface System, in *Proceedings of the First Conference of the Italian Association for Artificial Intelligence*, Trento, November 8-10 1989, pp.78-86.

Grisman, Ralph, TIPSTER text Phase II Architecture Design, New York Univ, 1996.

Marino, M., A Framework for the Development of Natural Language Grammars, in *Proceedings of the First Interantional Workshop on Parsing Technologies*, Carnegie-Mellon University, Pittsburgh, August 28-31 1989, pp. 350-360.

Marino, M., Bottom-up Parsing Extending Context-Freeness in a Process Grammar Processor, in

Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics (ACL), University of Pittsburgh, June 6-9 1990, pp. 299-306.

Peters W., Cunningham H., McCauley C., Bontcheva K., Wilks Y., Uniform Language Resources Access and Distribution, *First International Conference on Language Resources & Evaluation*, Granada, 1998, 13-17.

Prodanof I, Cappelli A., Moretti L., Carenini M., Moreschini P., Vanocchi M., A Grammar Development Environment for Reusable and Easily Customizable NL Applications, in *Proceedings of the First International Conference on Language Resources & Evaluation*, Granada, 1998, 611-618.