ELU: an Environment for Machine Translation

ISSCO
54 rte des Acacias
CH-1227 Geneva
Switzerland
e-mail: estival@divsun.unige.ch

1. Introduction

We present here ELU,[1] an environment for research in computational linguistics, and, in particular, machine translation. As its name indicates, ELU is one of the growing number of systems designed to employ unification as the principal computational mechanism; we shall mention below some respects in which it differs from other such systems.

In addition to investigations of MT techniques, ISSCO has been involved in research on evaluation of MT systems (cf. King and Falkedal, 1990).

The basic assumption underlying work on ELU is that the nature of (machine) translation is as shown in Figure 1 below – distinct grammars encoding language-specific information, and defining a relation between texts and representations, used in conjunction with a mechanism for transforming these representations:

Analysis	**Synthesis**
German text	French text
German Grammar	French Grammar
representations	representations

Transfer
mapping between representations

Figure 1: Translation

As will be apparent, the facilities provided in ELU have been strongly influenced by this view.

2. ELU

ELU exploits the parser and unifier of UD (Johnson and Rosner, 1989), but differs from that earlier system in the addition of a generator (Russell et al., 1990) and a transfer component (Estival et al., 1990; Russell et al., 1989).

ELU may be thought of as an enhanced PATR-II style environment for linguistic development. Throughout the system, including the transfer component, the syntax of the ELU language follows rather closely the PATR formalism which has become a standard for unification-based systems (Shieber, 1986). An environment of this kind is both a computational tool, and a formalism for representing linguistic knowledge, and we will consider ELU under those two aspects:

- as a computational work environment which provides
- a language for stating linguistic descriptions.

2.1. ELU: The Computational Tool

The characteristics of ELU as a computational tool for linguistic development and applications derive from its clear and well-defined formalism, which allows the representation of linguistic knowledge independently of any particular machine or application. Some properties of the system are worth emphasizing – it is:

- **machine independent**: ELU is a Common LISP program, and requires no specific interface, and
- **general purpose**: ELU is designed as a computational tool suitable for a large range of linguistic applications, such as:
  - the description of differing aspects of a particular language: lexical, morphological, syntactic, or semantic.
  - a specific task such as **parsing** or **generating** natural language texts or **transfer** between feature-structures (e.g. to map a range of logically equivalent representations to a canonical form, cf. Russell et al. (1989)).
  - **translation**, an application which includes the specific tasks mentioned above – parsing texts in the source language, manipulating the resulting representations by means of the transfer rules, and generating texts in the source language from the result of this manipulation.
- **user-friendly**:[2] This is obviously very important since one of the goals of the project is to provide

[1] "Environnement Linguistique d'Unification"

their analyses in the easiest and most natural way.

2.2. ELU: The Formalism

ELU is designed to offer the same formalism in all of its components, be it for synthesis, analysis or transfer. Such a formalism must have a clear transparent syntax and a declarative semantics. The basic properties of the ELU formalism are that it is **declarative** and it is **unification-based**. These two properties immediately give it certain advantages:

Declarativeness means that a description is a set of independent statements about the well-formed expressions of the language. This allows the system to be:

- **flexible**, permitting changes during development;
- **incrementable**, as the linguist need not be concerned with the order in which information is added or new phenomena accounted for;
- **reversible**: Grammar reversibility (or bidirectional grammars) is a highly desirable goal in the context of machine translation, i.e. using the same grammar as either source or target language description, a goal attested to in other centres working on MT (cf. Dymetman & Isabelle, 1988; Van Noord, to appear; Russell et al., 1990). And regardless of the application, the ability to generate with a grammar is extremely useful as a method of checking its adequacy. Transfer reversibility is a working hypothesis we are pursuing (Estival et al., 1990), but its consequences are harder to foresee; only experimentation will help us to (possibly) identify classes of natural language phenomena where the translation relation cannot be reversed.

Among the advantages deriving from a unification-based system, we first note that unification has become a central concept for a number of computational tools for linguistics (cf. Kay, 1983; Shieber, 1986; Carroll et al., 1988), and linguistic theories such as GPSG, HPSG and LFG (more generally, cf. Sag et al., 1986). More concretely, a unification-based formalism can be characterized by the following properties:

- **expressivity**, different types of analysis at different levels of abstraction,
- **uniformity**, across these different analyses or across grammatical components,
- **theory-neutrality** – the system doesn't impose any particular linguistic theory (however, it is particularly well-suited for the implementation of some of the fundamental properties of modern linguistic theories, i.e. lexicalism and the description of linguistic objects structured in terms of complex attribute-value pairs.

Similar to other unification-based systems the grammar is written as a set of context-free rules which define the structure of constituents. Constraining equations annotated on these rules define the combination of information, thus establishing the mapping between a complex feature structure and the text, which is simply a string of words. The lexicon in ELU, containing information about the morphological and syntactic processes that a given lexical item might undergo, is a separate component consisting of a 'base lexicon' and an optional morphological processor. In the morphological component, word forms are described as a concatenation of surface characters (organized as a finite-state automaton) and feature structures are assigned to a given sequence of characters or globally to a transition state. These word forms are associated to entries in the base lexicon where relational abstractions (see below) provide a concise way to express powerful lexical generalizations about the behaviour of the words.

Finally, the ELU formalism provides some extensions to the well-known unification-based systems (such as PATR-II and its derivatives) which make it more expressive and permit a more direct expression of linguistic generalizations. In particular:

- ELU provides a language for stating
  - **disjunction** over both atomic and general feature structures
  - atomic **negation**.
- ELU accepts **terms** (trees) and **lists** as attribute values in addition to feature structures.
- ELU allows direct manipulation of lists defined as feature values with primitives similar to 'append' and 'member'.
- ELU allows the linguist:
  - to define **variable path names**,
  - to **type feature structures** as a means of imposing linguistic constraints.
- ELU allows abstracting over sets of equations with **relational abstractions**. Relational abstractions, or macros, are similar to PATR templates, but they are a much more powerful tool, because they admit recursive and multiple definitions. They thus allow the user to state lexical, morphological and syntactic generalizations in a concise way.
- ELU allows **mapping** between representations.

The transfer component in ELU provides a formalism to state relations over sets of feature structures in order to e.g. transfer from a feature structure of one language to the feature structure of another (the output of which can serve as input to generation). Transfer rules associate the analysis of one feature structure with the synthesis of another; they may be thought of

[2] The ELU work environment offers a number of facilities to the user: various levels of debugging; tracing on individual rules and macros; incremental compilation; repetition of commands; graphic trace of rule application; scripting, and indication of system status at any time.

as a specialized variety of pattern-matching rule. They are local in nature, and permit the recursive analysis and synthesis of complex feature structures according to patterns specified in a format closely related to that of the other ELU components. The interpretation of transfer rules involves unification, albeit in a context which restricts it to the role of a structure-building operation. The rules specify information about two distinct feature structures (e.g. one for German and one for French) but they do not indicate the direction they will be applied in (i.e. no mention of source or target language); they are truly bidirectional, simply stating a relation that must hold between two feature structures. The direction they are to be applied in is specified when the rules are compiled in order to determine which set of feature structures will be matched against and which set are to be 'built', thus the same rules can be used in either direction.

3. Conclusion

The demo will present the various components of ELU – morphological analysis and generation, parsing, transfer and generation with grammars for French and German. The demonstration will be given by Dominique Estival.

References

Carroll, J, B. Boguraev, C. Grover and T. Briscoe (1988) "A Development Environment for Large Natural Language Grammars". Technical Report 127, Computer Laboratory, University of Cambridge.

Dymetman, M. and P. Isabelle (1988) "Reversible Logic Grammars for Machine Translation". *Proceedings of the Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Carnegie-Mellon University, Pittsburgh.

Estival, D. (1990) "Generating French with a Reversible Unification Grammar". *Proceedings of COLING 1990*, Helsinki.

Estival, D., A. Ballim, G. Russell and S. Warwick (1990) "A Syntax and Semantics for Feature Structure Transfer". Presented at the Third International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages, University of Texas at Austin.

Johnson, R. and M. Rosner (1989) "A rich environment for experimentation with unification grammars". *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, 182–189.

Kay, M. (1983). "Unification Grammar". Technical Report, Xerox Palo Alto Research Center.

King, M. and K. Falkedal (1990) "Using Test Suites in the Evaluation of Machine Translation Systems". *Proceedings of COLING 1990*, Helsinki.

Russell, G., A. Ballim, D. Estival and S. Warwick (1989) "A Language for the Statement of Binary Relations over Feature Structures". Internal report, ISSCO.

Russell, G., S. Warwick and J. Carroll (1990) "Asymmetry in Parsing and Generating with Unification Grammars". *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh.

Sag, I., R. Kaplan, L. Karttunen, M. Kay, C. Pollard, S. Shieber and A. Zaenen (1986) "Unification and Grammatical Theory". *Proceedings of the West Coast Conference on Formal Linguistics, vol.5*, U. of Washington. Stanford Linguistics Association.

Shieber, S. (1986) *An Introduction to Unification-based Approaches to Grammar.* Volume 4 of CSLI Lecture Notes.

Van Noord, G. (to appear) "Bottom Up Generation in Unification-Based Formalisms", in C. Mellish, R. Dale, and M. Zock (eds.) *Proceedings of the Second European Workshop on Natural Language Generation.*