Machine Translation Using Isomorphic UCGs

John L. BEAVEN*
Pete WHITELOCK

Department of Artificial Intelligence
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
Scotland

Abstract

This paper discusses the application of Unification Categorial Grammar (UCG) to the framework of Isomorphic Grammars for Machine Translation pioneered by Landsbergen. The Isomorphic Grammars approach to MT involves developing the grammars of the Source and Target languages in parallel, in order to ensure that SL and TL expressions which stand in the translation relation have isomorphic derivations. The principle advantage of this approach is that knowledge concerning translation equivalence of expressions may be directly exploited, obviating the need for answers to semantic questions that we do not yet have. Semantic and other information may still be incorporated, but as constraints on the translation relation, not as levels of textual representation.

After introducing this approach to MT system design, and the basics of monolingual UCG, we will show how the two can be integrated, and present an example from an implemented bi-directional English-Spanish fragment. Finally we will present some outstanding problems with the approach.

1 Background and Introduction

The aim of this paper is to explore how the linguistic theory known as Unification Categorial Grammar can be adapted to the general methodology of Machine Translation using Isomorphic Grammars, as pioneered by Landsbergen and others in the ROSETTA team [Landsbergen 87a, b].

UCG is one of several recent grammar formalisms [Calder et al. 86, Karttunen 86, Pollard 85] which are highly lexicalist, i.e. rules of syntactic combination are not a language-specific component of the grammar, but are very general in character, and combinatory information is primarily associated with lexical items.

Lexical items are represented by sets of feature-value pairs (where the values may be themselves sets of such pairs), and are combined by unification into objects of the same type. The language defined is thus the closure of the lexicon under the combinatory rules.

Landsbergen's work on Isomorphic Grammars follows Montague's approach of having a one-to-one correspondence between syntactic and semantic rules. A syntactic rule R_{SL} in the Source Language corresponds to a syntactic rule R_{TL} in the Target Language if and only if they are both associated with the same semantic operation R_{Sem}. The translation relation is then defined in a precise manner and it can be guaranteed that well-formed expressions in the Source Language are translatable, as there will be an expression in the Target Language that is derived in a corresponding way, and can therefore be considered as a possible translation of it.

*Supported by a studentship from the Science and Engineering Research Council.

According to Landsbergen, writing isomorphic grammars is a way of being explicit about the "tuning" of SL and TL grammars that is essential for reliable MT. The present paper is an attempt to adapt this approach to a type-driven mapping between syntax and semantics.

2 Isomorphic Grammars

We can recognise two basic relations of relevance in translation. namely, "possible translation" (which is symmetric), and "best translation" given the current context and much extra-linguistic knowledge (which is not symmetric). We take the task of the linguistic component of an MT system to be a correct and complete characterisation of the former, and will have nothing further to say about the latter.

An important problem that arises in an interlingual translation system is what Landsbergen [Landsbergen 87b] calls the "subset problem". If the analysis component generates a set L of interlingual expressions, and the generation component accepts a set L' of them, the only sentences that can be translated are those that correspond to expressions in the intersection $L \cap L'$. If the grammars of the source and target languages are written independently, there is no way of guaranteeing that they map the languages into the same subset.

The problem arises because a sufficiently powerful system of interlingual representation will contain an infinite number of logically equivalent expressions that represent a meaning of a given Source Language expression. Of course, the Source Language grammar will only associate a single one of these with a given SL expression. However, in the absence of specific tuning, this is not guaranteed to be the same one that the Target Language grammar associates with any of the translation equivalents.

Therefore, SL and TL grammars must be tuned to each other. This is not a problem specific to interlingual translation: in the transfer approach to MT system design, this tuning is effected by an explicit transfer module. The use of *Isomorphic Grammars* is another way of being explicit about this, tuning the grammars themselves rather than their inputs/outputs, which offers a greater possibility of bi-directionality than the transfer approach.

Landsbergen assumes the existence of compositional grammars for two languages, that is, grammars in which i) basic expressions correspond to semantic primitives and ii) each syntactic rule that builds up a complex linguistic expression from simpler ones is paired with a semantic rule that builds the meaning of the complex expression from the meanings of the simpler ones.

The tuning of grammars consists in ensuring that there is a basic expression in one grammar corresponding to each basic expression in the other, and that for each semantic rule there is a corresponding syntactic rule in each grammar. Two expressions are then considered possible translations of each other if they can be derived from corresponding basic expressions by applying corresponding syntactic rules. In other words, they are possible trans-

lations of each other if they are built from expressions having the same meaning, by using syntactic rules that perform the same semantic operations. Note the lack of directional specificity in this definition of the "possible translation" relation.

3 The (monolingual) UCG formalism

Many recent grammar formalisms [Shieber 86] represent linguistic objects as sets of attribute-value pairs. Values taken by these attributes may be atomic, variables, or they may themselves be sets of attribute-value pairs, so these objects may be thought of as Directed Acyclic Graphs (DAGs), in which directed arcs represent features, and the nodes at the end of these represent values. Such formalisms typically support re-entrancy, that is, they provide a mechanism for specifying that objects at the end of different paths are the same object.

Unification Categorial Grammar is such a formalism, which combines a categorial treatment of syntax with semantics similar to Kamp's Discourse Representation [Kamp 81]. Each linguistic expression licensed by the grammar corresponds to what is called a sign. A sign consists of four main entries or features, which are explained below:

1. **phonology** (orthography in the present case)
2. **syntax**
3. **semantics**
4. The **order** in which the terms combine.

Typical signs for the lexical entries *Mary* and *sings* may then look something like the following:

$$\begin{bmatrix} \textbf{phon:} & \text{"Mary"} \\ \textbf{synt:} & \text{np}\wedge\begin{bmatrix} \textbf{pers:} & \text{3rd} \\ \textbf{num:} & \text{sing} \\ \textbf{gen:} & \text{fem} \end{bmatrix} \\ \textbf{sem:} & \text{mary} \\ \textbf{ord:} & \text{Order} \end{bmatrix}$$

and

$$\begin{bmatrix} \textbf{phon:} & \text{sings} \\ \textbf{synt:} & \text{sent}\wedge\begin{bmatrix} \textbf{tense:} & \text{fin} \end{bmatrix}/\begin{bmatrix} \textbf{phon:} & \text{Phon} \\ \textbf{synt:} & \text{np}\wedge\begin{bmatrix} \textbf{pers:} & \text{3rd} \\ \textbf{num:} & \text{sing} \end{bmatrix} \\ \textbf{sem:} & \text{Sem} \\ \textbf{ord:} & \text{post} \end{bmatrix} \\ \textbf{sem:} & \text{[E][sings(E,Sem)]} \\ \textbf{ord:} & \text{Order} \end{bmatrix}$$

These are briefly explained below. Note that in the above example, as elsewhere, the Prolog-like convention is adopted that constants start with lower-case or are within quotes, and variables start with upper-case. Also, for the sake of simplicity in an introductory example, the first example above differs from the standard UCG practice of type-raising noun phrases, which follows Montague and others.

3.1 Syntax

There are 4 basic categories: nouns *(noun)*, sentences *(sent)*, noun phrases *(np)* and prepositional phrases *(pp)*. These may be further specified by features (such as **number**, **gender**, etc.). Features are indicated by the operator $\wedge$.

A category is either a basic category, or of the form A/B, where A is a category and B is a sign. Combination of signs is determined by the rule of function application, which allows a functor sign with syntax A/B to combine with an argument sign B', to give a sign like the functor sign but with syntax A. The combination is licensed if B and B' unify, and if the functor and argument signs appear in the order specified by the value of the **order** feature in B (if the **order** feature of an argument is *pre* its functor must precede it, and if it is *post* the functor follows it). The unification may further instantiate variables in the functor sign (in particular, the semantics). Although Function Application is the main combination rule, there are a few important unary rules, such as Gap Deletion, *pp*-insertion, and others. Unlike many other extended Categorial Grammars, UCG does not have Functional Composition, as a similar effect is achieved by the technique of Gap Threading, based on work by Johnson and Klein [Johnson and Klein 86]. However it is envisaged that a richer set of binary rules, and a reduction or elimination of unary rules, will be necessary if the Isomorphic Grammars approach is to be extended to typologically diverse languages.

3.2 Semantics

The semantic formalism used in UCG is similar to Kamp's DRT, but with a Davidsonian treatment of predicates. It is called InL (Indexed Language) and is described in [Zeevat 86]. A sentence like:

If a linguist owns a donkey, she writes about it

is represented in InL by:

$$[S1][[S2][[X]\text{linguist}(X), [Y]\text{donkey}(Y), [S2]\text{own}(S2,X,Y)]$$

$$\Rightarrow [E]\text{write\_about}(E,X,Y)]]$$

There is an important difference between InL and DRT: each formula introduces a discourse referent, or *index* ($S1$ and $S2$ above) which corresponds to the semantic object introduced by the formula. Since events, states etc. are primitive semantic objects, InL permits a first order treatment of modifiers.

Indices contain information about the sortal nature of the discourse referent in question. The sorts are coded into a subsumption lattice, and consist of bundles of features which may be unified. Unification ensures for instance that predicates have arguments of the right sort.

4 UCG and Isomorphic Grammars

The principle of Isomorphic Grammars is realised in UCG by means of bilingual signs. Bilingual rules, which combine bilingual signs, may be defined in terms of how monolingual rules combine the monolingual parts of the sign.

As was mentioned, monolingual UCG signs consist of four features: Phonology, Syntax, Semantics, and Order. A bilingual sign is merely a sign with top-level attributes **source** and **target** having monolingual signs as their values, and in which **source semantics** and **target semantics** share their value. Since translation must preserve semantics, this sharing of values is a necessary condition. In the general case, however, it is not sufficient (see section 5).

The Bilingual sign can easily be decomposed into, or built up from, a Source sign and a Target sign (having a common Semantics), by a Prolog predicate

```
decompose(Bilingual_Sign, Source_Sign, Target_Sign).
```

Combination of two monolingual signs is defined by two predicates:

```
source_combine(S1, S2, S).
target_combine(T1, T2, T).
```

which combine their first two arguments to give the third.

The crucial difference between these two predicates is as follows: **source_combine** requires that the **order** feature of S1 and S2 is consistent with the phonology of S, while **target_combine** ensures that the phonology of T is consistent with the **order** of T1 and T2. This enables differences in word order in the Source and Target Languages to be accounted for, as shown below.

The two monolingual modes of combination above are used to define bilingual combination through a predicate:

```
bilingual_combine(B1, B2, B):-
      decompose(B1, S1, T1),
      decompose(B2, S2, T2),
      source_combine(S1, S2, S),
      target_combine(T1, T2, T),
      decompose(B, S, T).
```

The way in which differences in word order are dealt with may be illustrated by the translatioin equivalence between an adjective-noun combination in English and a noun-adjective combination in Spanish. For the sake of simplicity, only the features for **phonology**, **syntax** and **order** are included.

The predicate `source_combine` allows two combinations:

(1) $\begin{bmatrix} \text{p:} & \text{W1} \\ \text{s:} & \text{A/B} \end{bmatrix} \begin{bmatrix} \text{p:} & \text{W2} \\ \text{s:} & \text{C} \\ \text{o:} & \text{pre} \end{bmatrix} \longrightarrow \begin{bmatrix} \text{p:} & \text{W1 W2} \\ \text{s:} & \text{A} \end{bmatrix}$

(2) $\begin{bmatrix} \text{p:} & \text{W1} \\ \text{s:} & \text{C} \\ \text{o:} & \text{post} \end{bmatrix} \begin{bmatrix} \text{p:} & \text{W2} \\ \text{s:} & \text{A/B} \end{bmatrix} \longrightarrow \begin{bmatrix} \text{p:} & \text{W1 W2} \\ \text{s:} & \text{A} \end{bmatrix}$

(where the active part of the functor sign unifies with the argument sign)

The predicate `target_combine`, on the other hand, allows the above two combinations, and in addition the two order-reversing ones:

(3) $\begin{bmatrix} \text{p:} & \text{W1} \\ \text{s:} & \text{A/B} \end{bmatrix} \begin{bmatrix} \text{p:} & \text{W2} \\ \text{s:} & \text{C} \\ \text{o:} & \text{post} \end{bmatrix} \longrightarrow \begin{bmatrix} \text{p:} & \text{W2 W1} \\ \text{s:} & \text{A} \end{bmatrix}$

(4) $\begin{bmatrix} \text{p:} & \text{W1} \\ \text{s:} & \text{C} \\ \text{o:} & \text{pre} \end{bmatrix} \begin{bmatrix} \text{p:} & \text{W2} \\ \text{s:} & \text{A/B} \end{bmatrix} \longrightarrow \begin{bmatrix} \text{p:} & \text{W2 W1} \\ \text{s:} & \text{A} \end{bmatrix}$

Let us then examine how the English expression *red book* gets translated into the Spanish *libro rojo*, in which the order of the adjective and noun are reversed.

The bilingual signs are:

$$\begin{bmatrix} \text{src:p:} & \text{red} \\ \text{tgt:p:} & \text{rojo} \\ \text{src:s:} & \text{noun/}\begin{bmatrix} \text{s:} & \text{noun} \\ \text{o:} & \text{pre} \end{bmatrix} \\ \text{tgt:s:} & \text{noun/}\begin{bmatrix} \text{s:} & \text{noun} \\ \text{o:} & \text{post} \end{bmatrix} \end{bmatrix}$$

and

$$\begin{bmatrix} \text{src:p:} & \text{book} \\ \text{tgt:p:} & \text{libro} \\ \text{src:s:} & \text{noun} \\ \text{tgt:s:} & \text{noun} \end{bmatrix}$$

These will get decomposed into their source and target constituents, which may only be combined using (1) and (3) above, respectively:

$$\begin{bmatrix} \text{p:} & \text{red} \\ \text{s:} & \text{noun/}\begin{bmatrix} \text{s:} & \text{noun} \\ \text{o:} & \text{pre} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{p:} & \text{book} \\ \text{s:} & \text{noun} \end{bmatrix} \xrightarrow{(1)} \begin{bmatrix} \text{p:} & \text{red book} \\ \text{s:} & \text{noun} \end{bmatrix}$$

$$\begin{bmatrix} \text{p:} & \text{rojo} \\ \text{s:} & \text{noun/}\begin{bmatrix} \text{s:} & \text{noun} \\ \text{o:} & \text{post} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{p:} & \text{libro} \\ \text{s:} & \text{noun} \end{bmatrix} \xrightarrow{(3)} \begin{bmatrix} \text{p:} & \text{libro rojo} \\ \text{s:} & \text{noun} \end{bmatrix}$$

Currently, we assume the existence of four bilingual signs corresponding to the English word *red*, since the Spanish adjective has four combinations of gender and number. Only that sign representing the contextually correct translation equivalence will be incorporated in the derivation. In a practical system, there would be a single bilingual sign whose Spanish component has disjunctive (or unspecified) values for gender and number, and the incorporation of this sign into the derivation will eliminate the disjunction (or bind the variables).

Unlike Landsbergen's approach, it is not necessary to specify that the rules which combine the SL and TL expressions must be the same. Because of the type-driven mapping between syntax and semantics, if two pairs of signs stand in the translation relation, then so will the pair of signs resulting from their combination, regardless of the rule used.

5 Current Difficulties

There are several important difficulties that remain unsolved. The first one is how to handle the differences in the freedom of word order in two languages. For instance, Spanish word order is relatively free compared to English. It conveys important stylistic information that should be captured in the translation, but which at present gets lost. Another aspect of the same problem is that we would like to be able to recognise all possible word orders in Spanish, without generating them all (as some are intelligible but sound awkward).

A possible solution to this could be to include some measure of the degree of "markedness" of a construction in each language. The translation process would attempt to keep the markedness of the two constructions as close as possible to each other. If the grammar specifies that Spanish sentences may be more "marked" than the English, the more marked would never be generated, though they could be analysed.

Another problem is how the set of basic bilingual signs is to be characterised. That the semantics of SL and TL signs unify is a necessary condition for them to stand in the relation of translation equivalence. It is however insufficient in two ways. First, it must be the case that there is no more specific sign in either language whose semantics unifies with that of the other language, and which is of similar markedness. Secondly, it must be the case that the semantics of the two signs will continue to unify regardless of the derivations into which the signs are incorporated. For instance, suppose that the English word *leg* is associated with the semantics [leg_of(X,Y)], and the Spanish word *pierna* with [leg_of(X,Y),human(Y)]. Although these semantic values do not contradict each other, they will if Y becomes bound to a non-human entity. In this case, the solution is clear - a further bilingual sign must be constructed in which English *leg* is paired with Spanish *pata*, having the semantics [leg_of(X,Y),not(human(Y))]. Then, either the derivation will eliminate one or the other equivalence, or both translations will be produced, which is the desired result.

It is possible that one monolingual component of a bilingual lexical sign will not be a basic expression in that language. Instead, it must be explicitly constructed in order to be paired with a basic expression in the other language. The unification-based semantics gives an indication of when such a sign-construction process must take place. The flexible categorial approach to the construction of constituents allows the non-standard categories needed to be built.

In a sense, all the hard work of this approach takes place at this point. See [Whitelock 88] for a discussion of the issues involved.

Finally, there is a cluster of problems that impinge on the question of computational efficiency. It seems unavoidable that certain bilingual signs will need to incorporate either discontinuous or null constituents, or both, from one or the other of the languages concerned.

6 Conclusion

This paper presents a view of MT that is based on the direct specification of a computable description of a recursive translation relation. We have proposed a system of simultaneous constraints placed on isomorphic derivation trees in SL and TL whose leaves are elements of a finite set of bilingual signs and whose internal nodes stand in a type-driven compositional relationship to their daughters. It is the combination of unification and categorial techniques that makes this idea particularly feasible. The non-standard constituents made available in a full categorial calculus enables isomorphic derivation trees to be built; the partiality of the signs and their combination by unification allows the expression of very precise constraints that both derivations must satisfy. The partiality of semantic representations is also crucial in determining the set of equivalences - the bilingual lexicon - that form the basis of the recursive translation relation.

There remain many problems with realising this approach in a practical system. However, we believe that there are significant advantages to be gained by a direct statement of the translation relation between two languages that is at once declarative, computable and linguistically well-founded.

References

[Calder et al. 86] Calder, J., Moens, M. and Zeevat, H. (1986) *A UCG interpreter.* ACORD Deliverable T2.6. Centre for Cognitive Science, Edinburgh University.

[Johnson and Klein 86] Johnson, M. and Klein, E. (1986) Discourse, anaphora and parsing. In *Proceedings of the 11th International Conference on Computational Linguistics and the 24th Annual Meeting of the Association for Computational Linguistics*, Institut fure Kommunikationsforschung und Phonetik, Bonn University, Bonn, August 1986.

[Kamp 81] Kamp, H. (1981) A theory of truth and semantic representation. In Groenendijk, J.A.G., Janssen, T.M.V. and Stokhof, M.B.J. (Eds) *Formal Methods in the Study of Language,* Vol 136, pp 227–322. Amsterdam: Mathematical Centre Tracts.

[Karttunen 86] Karttunen, L. (1986) Radical Lexicalism. CSLI–86–68, Centre for the Study of Language and Information, Stanford University, California.

[Landsbergen 87a] Landsbergen, J. (1987). Isomorphic Grammars and their Use in the ROSETTA Translation System. In King, M. (Ed) *Machine Translation Today: the State of the Art. Proceedings of the Third Lugano Tutorial, Lugano, Switzerland, 2–7 April 1984.* Edinburgh University Press.

[Landsbergen 87b] Landsbergen, J. (1987) Montague Grammar and Machine Translation. In Whitelock et al. (Eds). *Liguistic Theory and Computer Applications.* Academic Press.

[Pollard 85] Pollard, C. (1985). *Lectures on HPSG.* Unpublished lecture notes, CSLI, Stanford University.

[Shieber 86] Shieber, S. (1986) *An Introduction to Unification-based Approaches to Grammar.* Lecture Notes Number 4. Center for the Study of Language and Information, Stanford University.

[Whitelock 88] Whitelock, P. (1988) *The Organisation of a Bilingual Lexicon.* DAI Working Paper, Dept. of Artificial Intelligence, Univ. of Edinburgh.

[Zeevat 86] Zeevat, H. (1986). *A specification of InL.* Unpublished Internal ACORD Report. Centre for Cognitive Science, University of Edinburgh.

[Zeevat 87] Zeevat, H., Klein, E., and Calder, J. (1987). Unification Categorial Grammar. In Haddock, N., Klein, E. and Morrill, G. (Eds) (1987). *Working Papers in Cognitive Science, Vol. 1: Categorial Grammar, Unification Grammar and Parsing.* Centre for Cognitive Science, University of Edinburgh.