The MiMo2 Research System

Gertjan van Noord
Joke Dorrepaal
Pim van der Eijk
Maria Florenza
Louis des Tombe *

Research Institute for Language and Speech
State University Utrecht
Trans 10 3512 JK Utrecht
The Netherlands

Abstract

The MiMo2 translation system combines several leading ideas in the areas of linguistics, computation and translation. In the area of translation we follow the basic ideas of Landsbergen [26] by assuming that translation is symmetric; and combine these ideas with the advantages of a transfer approach. Computationally the system focuses on *computability* and *declarativity*. The linguistics of the system is based on a *lexicalistic* and *sign-based* approach to grammar.

1 Introduction

The MiMo2 system is based on a fundamental distinction between 'possible' and 'best' translation (Landsbergen [26]). Linguistically, a source text can have many possible translations, i.e. many target language texts that are equivalent in meaning. In practice, some interpretations will be more plausible than others, and some translations of some interpretations will often be preferred over others on various grounds, e.g. style. The MiMo2 system attempts to capture the notion 'linguistically possible translation', as will be clarified below. This leads to a system that produces plausible as well as implausible translations. In the future, the selection of the more plausible ones could be based on either interactivity or more machine intelligence.

*We were supported by the European Community and the NBBI through the Eurotra project.

The MiMo2 system differs from Landsbergen's work in two respects. First, it uses a transfer model, where the Rosetta system is based on an interlingua approach. Second, its rule formalism and linguistic theory are based on unification grammars and a lexicalist approach, where the Rosetta system uses M-grammars, a computationally tractable version of Montague grammar.

1.1 Views on Translation

1.1.1 Linguistically possible translation

Our criterion of linguistically possible translation is defined as follows:

- translation preserves logical meaning;
- moreover, it preserves as much as possible the way in which meaning is built compositionally;

The reasons for this are the following.

Machine translation finds its typical applications in non-literary texts, e.g. news bulletins, scientific articles, etc. It is obvious that in such texts, what is said about the world is the most important thing. For example, one would like the translation of a weather report or news bulletin to be true if and only if the original is true (even more so with airplane maintenance manuals).

However, the first criterion is too weak for practical applications. If for a given source text S there is a target language equivalent T, then a huge number of other target texts are equivalent too, like 'T and R' where R is a tautology. Similarly, all tautologies become translations of each other. A finer view of 'meaning' is necessary. New developments (e.g. property theory [10]) may lead to more interesting criteria but it is not clear at this moment how to apply them in practice. Therefore we require also some preservation of syntax. The way in which meanings of expressions are composed from the meanings of their parts is preserved wherever possible. To take an example from Lewis [27, p. 182] 'Snow is white or it isn't' differs in 'syntactic meaning' (though not in truth conditions) from 'Grass is green or it isn't' because the embedded sentences differ in truth conditions. The same idea has been applied by Landsbergen [26].

This view of translation is extremely poor. For example, it does not take world knowledge into account. Therefore, a sentence like 'the prime ministers discussed the situation of Iran in Moscow' is ambiguous in the MiMo2 system (even though the two meanings may sometimes lead to identical translations). Moreover, there are many other factors that could be taken into account in defining linguistically possible translation, e.g. preservation of style, (indirect) speech act, honorifics, etc. It is hoped (and expected) that an approach based on the poor view described can be useful as a basis for future richer views.

An important question of translation is whether there always is a meaning-preserving translation. It may be the case that there are meanings in one language that are not expressible at all in some other language (for some discussion cf. [21, 22]). It may even be the case that one cannot know whether the meaning expressed in two languages is the same (cf. [35]). These are important questions, but the MiMo2 system is irrelevant to them. It is concerned only with the case where the same meaning can be expressed in both languages. Our question is 'how to describe possible translations', not 'is translation possible'.

1.1.2 Symmetry

Since 'possible translation' is defined in terms of 'having the same syntactic meaning', the relation is symmetric. That is, for each pair of sentences S and T, S translates to T if and only if T translates to S. Because we preserve the syntactic structure only to a certain degree (wherever possible), we suspect that the relation is not transitive.

Given the assumption that linguistically possible translation is a symmetric relation, we have defined the computational model in such a way that it is reversible (see below).

1.1.3 Transfer model

The MiMo2 translation theory defines two types of differences in the way in which languages encode meaning: content words, and other syntactic means (word order, function words, morphology). It is based on the idea that content word meaning is difficult to represent in a universal way, and so it uses transfer to deal with this aspect of meaning.

A good example of the reason for having transfer is the difference between the Dutch word 'schimmel' and the English translation 'white horse'. The logical meaning is the same (in fact, this is not quite true, since a 'schimmel' cannot be a black horse painted white - we abstract away here from treating problems like this). But Dutch uses a primitive expression and English a complex one. Now if there is no transfer, there must be one 'pivot language' or 'interlingua' that serves as the point of communication between the two languages. A question is, whether the pivot language encodes the piece of meaning of this example as a primitive expression (e.g. 'schimmel'), or as a complex one (e.g. 'white horse'). In the first case, the English grammar must be complicated; in the second case, this applies to the Dutch grammar. Now in the bilingual case, this does not really matter, as the complication has to go somewhere anyway. But in a multilingual situation, each monolingual grammar will be complicated in this way by linguistic peculiarities of all the other languages. Think only of the English-Dutch translation pair 'snow-sneeuw' in a multilingual system that has Eskimo as one of its languages.

In the MiMo2 system, we can express the equivalence by a transfer rule like (for the actual notation see below):

$$schimmel \equiv (horse \wedge white)$$

In sum, an interlingual approach suffers from potential arbitrariness, and may complicate the overall system; but the issue is relevant only if one wants to keep open the possibility of a multilingual system.

1.2 Computation

Monolingual and bilingual knowledge is represented in a *declarative* way. Declarativity implies that the grammar writer does not have to worry about the actual processing of the linguistic knowledge he/she encodes, but only worries about the *logical* meaning of a grammar. Declarativeness has been argued for from a computational point of view because it implies that different compilers and interpreters may be applicable to the very same program. This has led to the bidirectional use of programs written in declarative grammar formalisms such as PATR and DCG [42, 31, 40]. Some recent developments are reported in [11, 54, 41, 49, 9, 43, 44, 50, 46].

We make a distinction between *symmetric* and *reversible*. We call a translation relation T *reversible* if T is *symmetric* and *computable*. Symmetry of the 'possible translation' relation has been argued for above. A relation $R \subseteq A \times B$ is called *computable* iff for a given $a \in A$ the set $\{b \in B | R(a, b)\}$ can be enumerated by some terminating procedure. We will return to this matter in section 2.1.

Reversible systems are preferable to nonreversible ones. The arguments in favour of using bidirectional grammars in NLP, such as those given in [3, 20] carry over to translation. Furthermore Isabelle [19] claims that reversible MT systems are to be preferred to others because in reversible MT systems a better understanding of the translation relation is achieved; such systems will eventually exhibit better practical performance. Monolingual grammars that are used only for analysis will often allow constructions that are in fact ungrammatical. As an example consider English auxiliaries. Suppose that the English auxiliaries are analyzed as verbs that take an obligatory *vp*-complement. Moreover each auxiliary may restrict the $vform$ (participle, infinite) of this complement. This allows the analysis of sentences such as 'John *will have been kissing* mary'. However, the possible *order* of English auxiliaries (eg. 'have' should precede 'be') is not accounted for and the analysis sketched above will for example allow sentences such as 'John will be having kissed Mary'. The strictness coming with a bidirectional grammar will be useful for analysis too, because strictness usually implies less local and global ambiguities.

1.3 Unification Linguistics

In MiMo2, grammars covering a basic subset of English, Dutch and Spanish have been developed. The linguistic theory embodied in these grammars is a variant of the emerging family of unification grammars (UG; see [40] for a general introduction), Head-Driven Phrase Structure Grammar (HPSG, [34]) being the initial source of inspiration. The usual implementation tool of these grammars is a member of the family of logic grammars, such as PATR. Two recent developments in the UG tradition adopted in MiMo2 are the sign-based approach and a strong lexical orientation.

In sign-based theories like HPSG and Unification Categorial Grammar (UCG, [55]), linguistic objects (grammar rules, lexical entries etc.) are described as *partial information structures* that express declarative and monotonic constraints on combinations of (possibly diverse) types of linguistic information [34, p 7]. As opposed to linguistic theories (such as transformational grammar) and NLP formalisms (such as the Eurotra's E-framework [5]) in which linguistic representations are sequentially transformed one into the other, sign-based grammars allow for interleaved processing of phonology, syntax, and semantics.

The second development in the UG tradition is a strong lexical orientation, which initiated with LFG [7]. MiMo2 follows HPSG [34, 38] in having small grammars with few but general rules and rich lexical entries. To minimize redundancies and to capture generalisations, it is possible to define macro's (cf. the 'let' definitions of [40], or the 'aliases' of [36]) to implement a lexical inheritance hierarchy ([13], [34] chapter 8). Furthermore, maintenance of large lexicons is facilitated by a separate lexical preprocessor, which is discussed elsewhere [48].

The lexicalist approach is partly motivated by the considerable reduction of grammar size it enabled us to achieve, e.g. by moving subcategorization frames to the lexicon, thereby eliminating the large number of phrase structure rules in earlier phrase structure grammar (GPSG [15]). This reduction is of relevance considering the maintenance complexity of large grammars. The possibility of separately defining linguistic principles, which can be called in grammar rules as macro's, reduces the grammar complexity.

Examples of principles are HPSG's SUBCAT-principle, which recursively realizes the head of the list of arguments, which represents the subcategorization frame (cf. functional application in Categorial Grammars), and the Head Feature Principle (the HPSG restatement of GPSG's Head Feature Convention in unification terms). These principles can be defined universally, so that they can be called from all grammar components. Use of these macro's sometimes allows for some modularity: the definition of the principle can be changed without changing any grammar rule.

An empirical motivation of the lexicalist approach is the huge amount of word-specific idiosyncracies. The combination of the sign-based lexicalist approach and the idea of the subcategorisation list enables linguists to describe the idiosyncratic character of idioms in the lexical entry of the head word of the idiom only, by directly specifying the argument on the subcatlist. Given the lexical entry in figure 1 the MiMo2 grammar will recognize the VP *kick the bucket* semantically as the one-place predicate *kick_bucket* (note that subjects are not on the subcat list in this approach).

Since the grammars are implemented in the PATR formalism, certain HPSG proposals, such as the ID/LP rule format and the obliqueness-hierarchy cannot

Figure 1: The lexical entry for 'kick'

```
kick →
⟨stem⟩ = kick
⟨syntax head cat⟩ = v
⟨syntax subcat first semantics pred⟩ = bucket
⟨syntax subcat rest⟩ = nil
⟨semantics pred⟩ = kick_bucket
⟨semantics arg1⟩ = ⟨syntax subject semantics⟩
```

be implemented directly. To a certain extent some proposed extensions could be simulated, as will be shown in section 3, but this is not in general the case.

The lexicalist approach can easily be extended to handle bilingual lexical idiosyncracies. This makes it fit well into the transfer-based view of translation described in section 1.1.3.

2 Overview of the formalism

2.1 Reversible Unification Grammars

A unification grammar defined in grammar formalisms such as DCG [31] and PATR [42] usually defines a relation between a string of words and some representation, sometimes called *logical form.* In sign-based approaches such as UCG [55] and HPSG [34], the string of words is not assigned a privileged status but is represented as the value of one of the attributes of a feature structure. In this approach a unification grammar defines a set of feature structures where each feature structure represents a pairing of a logical form and a string. Such a grammar can thus be seen as defining two relations, a relation $r \subseteq A \times B$ and an its converse relation $r^{-1} \subseteq B \times A$, where A and B are, for example, feature structures representing strings and feature structures representing logical forms. Together these relations define the symmetric relation $R \subseteq F \times F$, where F is the set of feature structures including A and B, and $R = r \cup r^{-1}$.

It is also possible to use unification grammars to define other symmetric relations between feature structures. In MiMo2 unification grammars are used to encode bilingual knowledge too: each (bilingual) unification grammar defines a symmetric transfer relation. Monolingual unification grammars define relations between strings and logical forms; bilingual grammars define relations between (language specific) logical forms.

In MiMo2 the translation relation between two natural languages is defined by a *series* of three unification grammars. Each unification grammar defines a symmetric relation, for example between Dutch strings and Dutch logical forms, or between Dutch logical forms and English logical forms. Moreover each of these relations is *computable*, and hence *reversible.* A relation $R \subseteq A \times B$ is called *computable* iff for a given $a \in A$ the set $\{b \in B | R(a,b)\}$ can be enumerated by some terminating procedure. Although in general unification grammars are not computable in this sense it is possible to constrain grammars to guarantee computability [32, 17, 51]. We say that the composition $r_1 \circ r_2$ of two relations r_1 and r_2 is $\{\langle x,y\rangle \mid \langle x,z\rangle \in r_1 \text{ and } \langle z,y\rangle \in r_2\}$. It is easy to see that if r_1 and r_2 are both reversible (symmetric and computable), $r_1 \circ r_2$ is also reversible.

For example the relation $R_{d,e}$ between Dutch and English strings is defined as the composition of the Dutch grammar in analysis direction, the Dutch-English bilingual grammar in the Dutch-to-English direction, and the English grammar in generation direction. Each translation relation that can be defined in MiMo2 is thus necessarily reversible.

2.2 Morphology

The monolingual components of the formalism thus consist of unification grammars, similar to PATR. Unlike PATR the terminal elements in the formalism are not defined in the lexicon, but orthographical, inflectional and morphological rules define the relation between the terminals and a lexicon of stems and affixes. For example, the word 'eaters' is analyzed into [eat,er,s] by the orthographical component. For the orthographical component we use a reversible two-level system [25, 6, 37]. Reversible inflectional rules relate to $[eat, er, s]$ a list of stems and affixes $[eat, er]$ with the feature structure $[cat : n, number : plur]$. Inflection is defined by a formalism comparable to the paradigmatic approach of [8]. Morphological analysis is based on a separate reversible unification grammar in which derivational processes and compounding can be defined [4]. For example, $[eat, er] + [cat : n, number : plur]$ could be analysed as $[cat : n, number : plur, semantics : [pred : er, arg1 : eat], ntype : agentive]$. Note that the separation of inflectional rules and compound/derivation rules implements a type of 'level' theory defended by e.g. [2].

2.3 Implementation

The unification grammars defined by the user in a PATR like style are compiled into Prolog (an extension to the compilation described in [18]), to enable an efficient implementation of parsing, generation and transfer. The parser is a 'left-corner' parser augmented with a well formed substring table and a reachability table [30, 28]. Both techniques are optimized by using a set of 'restricted' features [39]. The generator of Mimo2 is a member of the generation family described in [49, 43, 50, 44]. Transfer is implemented as a top-down backtrack search procedure.

3 Illustration

3.1 Linguistics in MiMo2

In MiMo2 grammar fragments of English, Dutch, and Spanish have been implemented. As testing and development methodology, the fragments have been defined to cover a specific text type, the one of international news items of teletext. As is well-known from studies on sublanguages (e.g. [24]), texts from a restricted domain show a greater 'adherence to systematic usage' than the standard language, which is a useful restriction in the development of accurate grammars. As it happens, the text type of teletext is rather close to the standards of written language as traditionally studied (mainly grammatical declarative sentences, little jargon and ellipsis etc.). However, it also has some frequent constructions which are highly restricted in standard language, such as a restricted type of apposition of proper nouns (*president Bush* vs. $*$*dissident Ajrikjan*), which has been analysed as well.

Despite the relatively 'standard' character of the text type, much pioneering work in linguistics was necessary due to the fact that there is not yet a large literature on language description using unification grammars. This is especially true for languages other than English. Detailed analyses of non-trivial phenomena will be given in the subsequent section.

3.1.1 Dutch Verb Clusters

As a very Dutch example, we describe cross serial dependencies, a phenomenon elaborately discussed by [12] and many others since. This phenomenon deals with the type of construction that contains verbs triggering V-raising, thus obtaining sequences of NPs and Vs with crossing dependencies. In the example a subordinate clause is used to circumvent interaction with 'Verb Second' in main clauses (see next section). The crossing dependencies are indicated by indices.

dat de voorzitter$_i$ de ministers$_j$ het voorstel$_j$ hoorde$_i$ bespreken$_j$
that the chairman the minister the proposal heard discuss (lit.)
that the chairman heard the ministers discuss the proposal

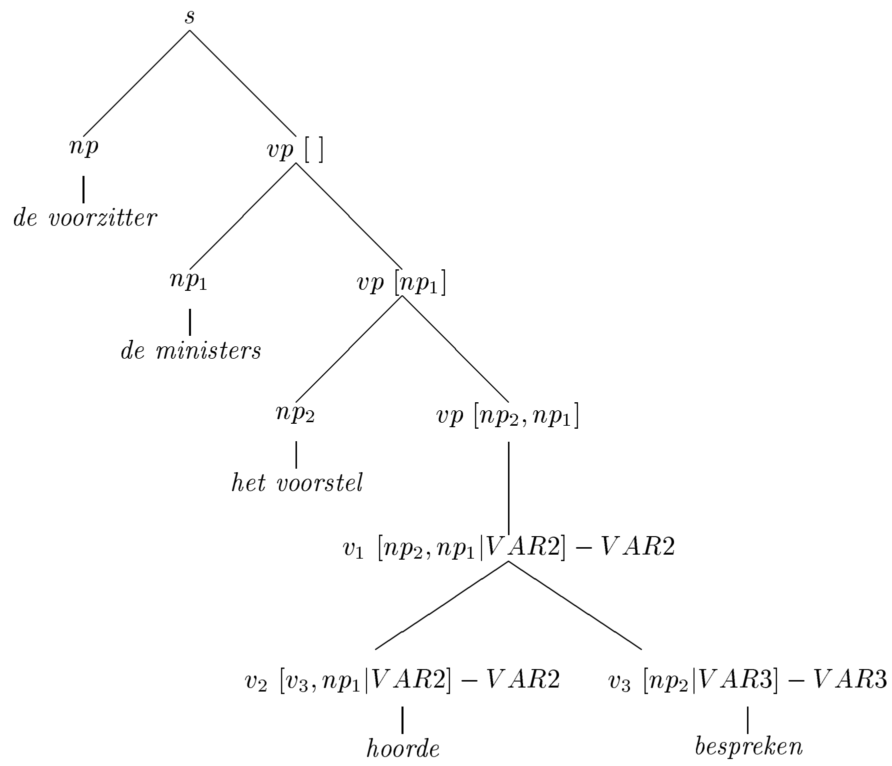
Following the analyses of [29, 45] in Categorial Grammar, we can view the sentence-final verb cluster as a complex predicate derived by composition of its constituent verbs. Taking subcatlists in Phrase Structure Grammar to be the equivalent of the (curried) functions of Categorial Grammar (as discussed in [33]), the composition operation can be implemented by appending the subcatlists of the clustered verbs. More precisely, the subcatlist of the first verb must be appended to the second, which as a whole must be appended to the third, if present, etc. This in order to guarantee that the subcategorized for arguments of the lower verb can be found higher up.

It would be possible to add to the formalism an 'append' operation as extra primitive operation on feature structures (this is the approach taken in [34, p. 48]), but the effect can be achieved directly with difference lists ([30, pp. 126-128]). This is done in the following annotated rule.

$v_1 \rightarrow v_2\ v_3$
$\langle v_2$ `subcat in first`$\rangle$ = v_3
$\langle v_2$ `subcat in rest`$\rangle$ = $\langle v_3$ `subcat out`$\rangle$
$\langle v_1$ `subcat out`$\rangle$ = $\langle v_2$ `subcat out`$\rangle$
$\langle v_1$ `subcat in`$\rangle$ = $\langle v_3$ `subcat in`$\rangle$

Figure 2 illustrates this rule in terms of Prolog lists.

Figure 2: Dutch verb raising



In the lexicon the subcatlists of verbs all show a tail variable ('$\langle$subcat out$\rangle$') unified with the end of the list. This technique makes it possible to have an unlimited number of raising-verbs, the subcatlists of which will all be appended to yield the right assignment of argument to verbs.

3.1.2 Spanish Verb Movement

The Spanish module of MiMo2 captures the following generalizations about Spanish syntax. The first position is optionally filled by a topicalised constituent (e.g. PP, AdvP, NP). Furthermore all objects and the subject can follow the verb cluster; moreover the subject can precede (part of) the verb cluster. The examples of possible orders (modulo topicalization) are given in the following sentences:

Juan da el libro a Maria
(Juan gives the book to Maria)
Juan da a Maria el libro
Da Juan a Maria el libro
Da Juan el libro a Maria
Da el libro Juan a Maria
Da el libro a Maria Juan
Da a Maria el libro Juan
Da a Maria Juan el libro

To get the SV order as well as the VS order, we use the following two rules to place the subject in the right position:

```
VP --> Subj VP
VP --> VP Subj
```

Objects are selected with the following rule:

```
VP --> VP Obj
```

where the SUBCAT principle applies in the standard way. Free order among subcategorised complements is a problem that can be solved lexically or by morphological rules. Topicalization is handled by gap-threading (as described for relative clauses in the next subsection).

As it stands the rules do not yet deal with the possibility for part of the verb cluster to precede the subject, while the other part follows it:

[*Ha estado*] María *buscando* petroleo
(Has been Maria searching oil)

This verb cluster can include the negation marker and other types of clitics and a possible perfective auxiliary (haber) before the progressive auxiliary 'estar'. The auxiliaries 'ser' (to be) and 'haber' (to have) can not immediately precede the subject.

Verb clusters are built with the following rules:

```
VP --> V1
V1 --> V1 V
V1 --> clitic V1
```

To get the order where the subject is preceded by part of a verb cluster (a V1 consituent) we use a threading analysis:

```
VP --> V1 VP/V1
V1/V1 --> []
```

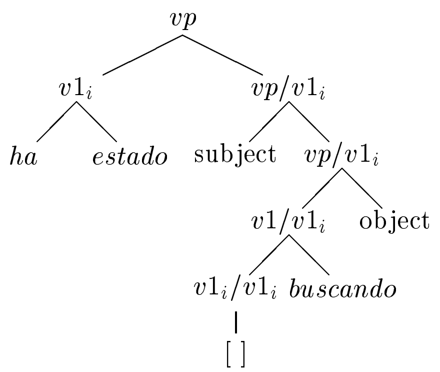
This analysis facilitates a compositional semantics of the verb cluster. Note that this analysis handles cases where clitics are part of the V1:

[*No se ha estado*] Juan equivocando

(Not himself has been Juan mistaking)

A simplified derivation tree of a sentence with two auxiliaries is shown in Figure 3.

Figure 3: Spanish Verb Movement



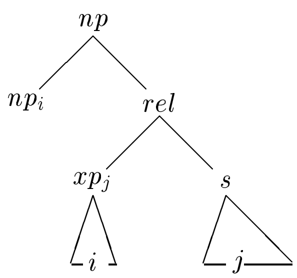
At the moment we are looking whether we can extend the verb threading analysis to verbs like modals and other aspectuals, which according to [47] have the same distribution as 'estar'.

In Dutch root sentences a similar construction exists, known as 'Verb Second'. Although this verb 'movement' is somewhat different from Spanish (it is not lexically restricted, it can be over more constituents, and is restricted to finite verbs), the implementation is very similar to the Spanish threading analysis as described here.

3.1.3 Relative Clauses

The implementation of relative clauses, both restrictive and non-restrictive, covers NP- and PP-extraction. The implementation differs from the analysis pro-

Figure 4: Relative clauses



posed in [34, p. 77] in two respects. First, [34] argues that the relation between the antecedent noun (phrase) and the relative pronoun, marked by the subscript i above, is the sharing of semantic information only. This may suffice for English but it does not for Dutch. Dutch requires agreement of syntactic gender as well, as the following examples show:

het meisje dat/∗die ik gisteren ontmoette
the (neuter) girl that I met yesterday

de jongen die/∗dat ik gisteren ontmoette
the (nonneuter) boy that I met yesterday

It shows that the information shared between antecedent and relative pronoun should not only consist of semantic but also of syntactic information.

The second difference between the HPSG-analysis and ours concerns unbounded dependencies like j in Figure 4, topicalisation and wh-movement. The SLASH feature employed in HPSG brings about full unification between the antecedent and the gap, thus identifying gap and antecedent entirely. It is however not desirable to have full unification between gap and antecedent, as is shown by the following data:

the man whom/who/[ ]/that I spoke to

the man to whom/∗who/∗[ ]/∗that I spoke

The differences between the pied-piping examples and the data that show preposition stranding can only be accounted for if we distinguish between gaps and overt antecedents. In our analysis, the relation between gap and antecedent is brought about by the unification of head features only. This allows gaps and antecedents to be different in other respects.

As to structural differences between English and Dutch w.r.t. relative clauses, the latter only allows preposition stranding in some special cases whereas English allows it quite freely. P-stranding in Dutch is only allowed when the complement of the preposition is a non-human pronoun, in which case it obligatorily appears in the so-called R-form and precedes the P of which it is a complement (cf. [53]).

de katedraal *waarnaar* Marie kijkt / *waar* Marie *naar* kijkt
the cathedral which_at Mary is looking / which Mary at is looking (lit.)
the cathedral at which Mary is looking / which Mary is looking at

de jongens *naar wie* Marie kijkt / ∗ *wie* marie *naar* kijkt
the boys at whom Mary is looking / whom Mary at is looking (lit.)
the boys at whom Mary is looking / whom Mary is looking at

Unification of gap and antecedent in unbounded dependencies is achieved by means of the gap-threading technique [30]. The method allows a straightforward analysis of the differences described. The Dutch PP consisting of an R-pronoun and a preposition allows the pronoun to be gapped and unified with an antecedent. The same holds for English PPs. Dutch PPs consisting of a P and a non-R pronoun are islands to gap-threading. The P-complement cannot be a gap, hence no preposition stranding will occur in these cases. The PP can of course be gapped in its entirety, yielding the pied-piping variant.

3.1.4 The Analysis of PPs

Prepositional phrases (which are extremely frequent both in general and in the teletext text type) pose at least three major problems. First, it is necessary to determine the semantic role of the PP, and therefore to find the correct reading of the head preposition, which is in general highly polysemous (frequent English prepositions like *after*, *at*, *in* and *of* are assigned, respectively, 9, 10, 21 and 14 readings in a medium-size dictionary like Longman's [1]). Second, it is necessary to deal with collocations to explain the ungrammaticality of phrases like ∗*at London* and ∗*in Christmas* although *London* can be combined with locative and *Christmas* with temporal prepositions. Finally, it is desirable to constrain PP-attachment to reduce structural ambiguity.

The first two problems can be dealt with jointly by explicitly marking all nouns directly for the preposition readings they can combine with by means of a collocational feature. This feature is percolated to the NP node as a result of the Head Feature Principle. The compositionality of the assignment of the semantic role to the collocation is expressed in the PP grammar rule by unification of the feature of the NP with the specification of the preposition, which filters disallowed combinations. The semantic role of the PP is then unified with the semantic role feature which is assigned lexically to the preposition.

This filter requires adding extra information to the nouns in the lexicon. In practice coding effort can be reduced considerably by defining macro's, the names of which are derived from thesauric classes that share distributional properties. For instance, in some languages names of countries all combine with the same preposition.

As an implementation note, it should be remarked that the strategy requires disjunction of values in the formalism, which is impossible in formalisms like PATR. However, if the value sets are finite, as in the case at hand, disjunction of values can be simulated by what is called a 'perverse' method in [14].

The attachment problem can be dealt with similarly. The semantic role of the PP is determined as described. The role of other modifiers, such as AdvPs, is assumed to be assigned lexically. By marking modified constituents such as VPs for the semantic roles they can be modified by, a similar filter as in the PP rule can be applied here.

It turned out that in our corpus linguistic restrictions only play a minor role in the reduction of PP-attachment ambiguities. As an example, it was already referred to in section 1.1.1. that geographic rather than linguistic knowledge is needed to determine the correct analysis of a sequence of locative PPs. We do not know whether this holds for other subject domains as well.

3.2 Transfer examples

A unification grammar defines the transfer relation between logical forms of two languages. Like in generation, the 'input attribute' is a logical form. Instead of strings, logical forms of the target language are generated.

For example the input to transfer may be a feature structure such as figure 5 for a sentence such as 'The army opens fire at the civilians' where *gb* contains the English logical form and *sp* the Spanish logical form [1].

The bilingual grammar will apply its rules, testing after each application whether the value of the attribute *gb* subsumes the input feature structure. The value of the attribute *sp* will gradually be instantiated. At the end of the process, the system will test whether the input feature structure subsumes the value of the attribute *gb*. If this is the case, then the value of *sp* will be

[1] It is assumed that monolingual analysis already has analysed 'open fire at' as an idiomatic construction: the value for *pred* will thus be an atomic identifier.

Figure 5: A possible input for transfer

$$\begin{bmatrix} gb = \begin{bmatrix} pred = open\_fire\_at \\ arg1 = \begin{bmatrix} pred = army \\ number = sg \end{bmatrix} \\ arg2 = \begin{bmatrix} pred = civilian \\ number = pl \end{bmatrix} \end{bmatrix} \\ sp = \end{bmatrix}$$

Figure 6: A simple rule

```
0 → 1 2 3
⟨0 gb pred⟩ = ⟨1 gb⟩ ⟨0 gb arg1⟩ = ⟨2 gb⟩
⟨0 gb arg2⟩ = ⟨3 gb⟩ ⟨0 sp pred⟩ = ⟨1 sp⟩
⟨0 sp arg1⟩ = ⟨2 sp⟩ ⟨0 sp arg2⟩ = ⟨3 sp⟩
```

considered the output of transfer. An example of a simple rule in PATR notation is given in Figure 6. The integers $0-3$ are names of feature structures, where 0 will normally be used to represent the mother node and $1 \ldots n$ represent the daughter nodes. Application of this rule to the feature structure of Figure 4 results in the three instantiations in Figure 7.

Figure 7: Three instantiations

$$\begin{bmatrix} gb = open\_fire\_at \end{bmatrix} \begin{bmatrix} gb = \begin{bmatrix} pred = civilian \\ number = pl \end{bmatrix} \end{bmatrix} \begin{bmatrix} gb = \begin{bmatrix} pred = army \\ number = sg \end{bmatrix} \end{bmatrix}$$

An example of the rule for the first daughter will be a lexical entry and may look as in Figure 8. The simple English expression 'army' has to be translated as a complex expression in Spanish: 'fuerza militar'. The rule will look as in Figure 9 where it is assumed that the construction is analysed in Spanish as an ordinary noun-adjective construction, and where the logical form of the adjective takes the logical form of the noun as its argument. The translation for 'civilian' is defined in a similar rule (although the translation of 'number' is different). Note that this example of complex transfer is similar to the famous 'schimmel - white horse' cases. As a result of the rule applications the value of the *sp* attribute in Figure 5 will get instantiated to the feature structure in Figure 10, from which the generator generates the string 'La fuerza militar rompio el fuego a la poblacion civil'.

Figure 8: A lexical entry

```
0 →
⟨0 gb⟩ = open_fire_at ⟨0 sp⟩ = romper_el_fuego_a
```

Figure 9: A rule for 'fuerza militar'

```
0 →
⟨0 gb pred⟩ = army         ⟨0 sp pred pred⟩ = militar
⟨0 sp arg1 pred⟩ = fuerza ⟨0 sp arg1 number⟩ = ⟨0 gb number⟩
```

For some transfer equivalences a rule such as in Figure 6 will be too simplistic. The rule in Figure 11 is used to translate a logical form such as $like(A, B)$ into the Dutch equivalent $bevallen(t(B), t(A))$, as in for example 'Minister Kok likes the reforms' vs. 'De hervormingen bevallen minister Kok'. Note that the attribute *nl* contains the Dutch logical form.

The logical forms that are encoded in MiMo2 are more complex than in the foregoing examples. For example attributes for *tmp* and *voice* may be present to represent information about voice, tense and aspect. The example in Figure 11 shows that it is sometimes necessary to alter the value of 'voice'. The Dutch logical form $heten(A, B)$ with $voice = active$ is related to $call(\_, t(A), t(B))$ with $voice = passive$. This rule may look as in Figure 12.

Figure 10: The target feature structure after transfer

$$\begin{bmatrix} pred = romper\_el\_fuego\_a \\ arg1 = \begin{bmatrix} pred = \begin{bmatrix} pred = militar \end{bmatrix} \\ arg1 = \begin{bmatrix} pred = fuerza \\ number = sg \end{bmatrix} \end{bmatrix} \\ arg2 = \begin{bmatrix} pred = \begin{bmatrix} pred = civil \end{bmatrix} \\ arg1 = \begin{bmatrix} pred = poblacion \\ number = sg \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

This approach is not entirely without problems. It seems that some redundancies might be inevitable with the architecture proposed in section 2. A monolingual grammar defines a relation between strings and logical forms. It thus defines possible logical forms as well. Similarly, a bilingual unification grammar will have to define possible logical forms. The notion 'possible logical

Figure 11: Like - bevallen

```
0 → 1 2
⟨0 gb pred⟩ = like      ⟨0 nl pred⟩ = bevallen
⟨0 gb arg1⟩ = ⟨1 gb⟩    ⟨0 gb arg2⟩ = ⟨2 gb⟩
⟨0 nl arg1⟩ = ⟨2 nl⟩    ⟨0 nl arg2⟩ = ⟨1 nl⟩
```

Figure 12: Heten - is called

```
0 → 1 2
⟨0 gb pred⟩ = call       ⟨0 nl pred⟩ = heten
⟨0 gb voice⟩ = passive  ⟨0 nl voice⟩ = active
⟨0 gb arg2⟩ = ⟨1 gb⟩     ⟨0 nl arg1⟩ = ⟨1 nl⟩
⟨0 gb arg3⟩ = ⟨2 gb⟩     ⟨0 nl arg2⟩ = ⟨2 nl⟩
⟨0 nl tmp⟩ = ⟨0 gb tmp⟩
```

form' will thus have to be defined in two (or more in case of a multilingual system) places. Note that this problem is a problem for all transfer systems; it is not restricted to our specific implementation of a transfer system (although the examples where the problem shows up may be specific to our grammars).

Some constructions such as control verbs and relative clauses may be represented using reentrancies; for example 'the soldiers tried to shoot the president' may be represented by a feature structure where the first argument of 'try' is reentrant with the first argument of 'shoot', cf. Figure 13. The translation of such logical forms to Dutch equivalents can be defined as in rule 14. However, it is not clear that such reentrancies are always as local as in this case; if the reentrancies can be further away the transfer grammar will have to be complicated (eg. by a threading mechanism) to be able to translate such constructions, although a transfer writer would prefer the possibility to state that 'reentrancies

Figure 13: A logical form containing reentrancy

$$\left[\ gb = \begin{bmatrix} pred = try \\ arg1 = \boxed{1}\begin{bmatrix} pred = soldier \\ number = pl \end{bmatrix} \\ arg2 = \begin{bmatrix} pred = shoot \\ arg1 = \boxed{1} \\ arg2 = \begin{bmatrix} pred = president \\ number = sg \end{bmatrix} \end{bmatrix} \end{bmatrix} \ \right]$$

Figure 14: Translating reentrancy

```
0 → 1 2 3
⟨0 gb pred⟩ = ⟨1 gb⟩           ⟨0 nl pred⟩ = ⟨1 sp⟩
⟨0 gb arg1⟩ = ⟨0 gb arg2 arg1⟩ ⟨0 nl arg1⟩ = ⟨0 nl arg2 arg1⟩
⟨0 gb arg1⟩ = ⟨2 gb⟩           ⟨0 nl arg1⟩ = ⟨2 sp⟩
⟨0 gb arg2⟩ = ⟨3 gb⟩           ⟨0 nl arg2⟩ = ⟨3 sp⟩
```

should be copied over' (cf. [52]).

4 Concluding Remarks

MiMo2 is an experiment in the application of reversible unification grammars to MT. Every translation relation is uniformly defined by a series of three unification grammars, following Landsbergen's hypothesis [26] that translation is a symmetric relation. Computationally, the system is to be characterized as *reversible* and *declarative*.

The current prototype translates a significant subset of news text sentences between the languages Dutch, English and Spanish. Current research includes quantification, modification, coordination and verb/noun collocations. Extensive research is furthermore required on topics related to discourse. The sentence is the unit of translation in the current prototype, which leaves us with problems like pronoun resolution.

References

[1] *Longman Dictionary of Contemporary English*. Longman House, 1987. Second Edition.

[2] M. R. Allen. *Morphological Investigations*. PhD thesis, University of Connecticut, 1978.

[3] Douglas E. Appelt. Bidirectional grammars and the design of natural language generation systems. In *Theoretical Issues in Natural Language Processing 3*, pages 206–212, 1987.

[4] John Bear. A morphological recognizer with syntactic and phonological rules. In *Proceedings of the 11th International Conference on Computational Linguistics*, 1986.

[5] Annelise Bech and Anders Nygaard. The E-Framework: A formalism for natural language processing. In *Proceedings of the 12th International Conference on Computational Linguistics*, 1988.

[6] Alan Black, Graeme Ritchie, Steve Pulman, and Graham Russell. Formalisms for morphographemic description. In *Third Conference of the European Chapter of the Association for Computational Linguistics*, 1987.

[7] Joan Bresnan, editor. *The Mental Representation of Grammatical Relations.* MIT Press, 1982.

[8] Jonathan Calder. Paradigmatic morphology. In *Fourth Conference of the European Chapter of the Association for Computational Linguistics*, 1989.

[9] Jonathan Calder, Mike Reape, and Henk Zeevat. An algorithm for generation in unification categorial grammar. In *Fourth Conference of the European Chapter of the Association for Computational Linguistics*, pages 233–240, 1989.

[10] Gennaro Chierchia and Raymond Turner. Semantics and property theory. *Linguistics and Philosophy*, 11:261–302, 1988.

[11] Marc Dymetman and Pierre Isabelle. Reversible logic grammars for machine translation. In *Proceedings of the Second International Conference on Theoretical and Methodological issues in Machine Translation of Natural Languages*, 1988.

[12] Arnold Evers. *The Transformational Cycle in Dutch and German.* PhD thesis, Rijksuniversiteit Utrecht, 1975.

[13] D. Flickinger, C. Pollard, and T. Wasow. Structure sharing in lexical representation. In *23th Annual Meeting of the Association for Computational Linguistics*, 1985.

[14] G. Gazdar, G.K. Pullum, R. Carpenter, E. Klein, T.E. Hukari, and R.D. Levine. Category structures. *Computational Linguistics*, 14(1), 1988.

[15] Gerald Gazdar, Ewan Klein, Geoffrey Pullum, and Ivan Sag. *Generalized Phrase Structure Grammar.* Blackwell, 1985.

[16] Barbara Grosz, Karen Sparck Jones, and Bonny Lynn Webber, editors. *Readings in Natural Language Processing.* Morgan Kaufmann, 1986.

[17] Andrew Haas. A generalization of the offline parsable grammars. In *27th Annual Meeting of the Association for Computational Linguistics*, 1989.

[18] Susan Hirsch. P-PATR: A compiler for unification-based grammars. In Veronique Dahl and Patrick Saint-Dizier, editors, *Natural Language Understanding and Logic Programming II.* North Holland, 1988.

[19] Pierre Isabelle. Towards reversible MT systems. In *MT Summit II*, 1989.

[20] Paul S. Jacobs. Achieving bidirectionality. In *Proceedings of the 12th International Conference on Computational Linguistics*, 1988.

[21] J.Katz. Effability and translation. In Guenthner and Guenthner-Reutter, editors, *Meaning and Translation.* Duckworth, 1978.

[22] E. Keenan. Some logical problems in translation. In Guenthner and Guenthner-Reutter, editors, *Meaning and Translation.* Duckworth, 1978.

[23] Margaret King, editor. *Machine Translation, the State of the Art.* Edinburgh University Press, 1987.

[24] R. I. Kittredge. The significance of sublanguage for automatic translation. In Sergei Nirenburg, editor, *Machine Translation, Theoretical and Methodological Issues.* MIT press, 1987.

[25] Kimmo Koskenniemi. Two-level morphology: a general computational model for word-form recognition and production. Technical Report 11, Department of General Linguistics, University of Helsinki, 1983.

[26] Jan Landsbergen. Isomorphic grammars and their use in the Rosetta translation system, 1984. Paper presented at the tutorial on Machine Translation, Lugano 1984, Also appears in [23].

[27] D. Lewis. General semantics. In D. Davidson and G.Herman, editors, *Semantics of Natural Language*. Reidel Dordrecht, 1972.

[28] Y. Matsumoto, H. Tanaka, H. Hirakawa, H. Miyoshi, and H. Yasukawa. BUP: a bottom up parser embedded in prolog. *New Generation Computing*, 1(2), 1983.

[29] Michael Moortgat. A fregean restriction on meta-rules. In *Proceedings of NELS 14*, 1984.

[30] Fernando C.N. Pereira and Stuart M. Shieber. *Prolog and Natural Language Analysis*. Center for the Study of Language and Information Stanford, 1987.

[31] Fernando C.N. Pereira and David Warren. Definite clause grammars for language analysis - a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13, 1980. reprinted in [16].

[32] Fernando C.N. Pereira and David Warren. Parsing as deduction. In *21st Annual Meeting of the Association for Computational Linguistics*, 1983.

[33] Carl Pollard. Categorial grammar and phrase structure grammar: An excursion on the syntax-semantics frontier. In R. T. Oehrle, Emmon Bach, and D. Wheeler, editors, *Categorial Grammars and Natural Language Structures*. Reidel, 1988.

[34] Carl Pollard and Ivan Sag. *Information Based Syntax and Semantics*. Center for the Study of Language and Information Stanford, 1987.

[35] W. V. Quine. *Word and Object*. MIT Press, 1960.

[36] Graeme Ritchie, Steve Pulman, Alan Black, and Graham Russel. A computational framework for lexical description. *Computational Linguistics*, 13(3-4), 1987.

[37] Herbert Ruessink. Two-level formalisms. *Working Papers in Natural Language Processing, Katholieke Universiteit Leuven, Stichting Taaltechnologie Utrecht*, 5, 1989.

[38] Ivan Sag. Grammatical hierarchy and linear precedence. *Syntax and Semantics*, 20, 1987. special issue Discontinuous Constituency.

[39] Stuart M. Shieber. Using restriction to extend parsing algorithms for complex-feature-based formalisms. In *23th Annual Meeting of the Association for Computational Linguistics*, 1985.

[40] Stuart M. Shieber. *Introduction to Unification-Based Approaches to Grammar*. Center for the Study of Language and Information Stanford, 1986.

[41] Stuart M. Shieber. A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics*, 1988.

[42] Stuart M. Shieber, Hans Uszkoreit, Fernando C.N. Pereira, J. Robinson, and M. Tyson. The formalism and implementation of PATR-II. In B. J. Grosz and M. E. Stickel, editors, *Research on Interactive Acquisition and Use of Knowledge*. SRI report, 1983.

[43] Stuart M. Shieber, Gertjan van Noord, Robert C. Moore, and Fernando C.N. Pereira. A semantic-head-driven generation algorithm for unification based formalisms. In *27th Annual Meeting of the Association for Computational Linguistics*, 1989.

[44] Stuart M. Shieber, Gertjan van Noord, Robert C. Moore, and Fernando C.N. Pereira. Semantic-head-driven generation. *Computational Linguistics*, 1990. To appear.

[45] Marc Steedman. Dependency and coordination in the grammar of dutch and english. *Language*, 61, 1985.

[46] Tomek Strzalkowski. Automated inversion of a unification parser into a unification generator. Technical report, Courant Institute of Mathematical Sciences, New York University, 1989. technical report 465.

[47] Esther Torrego. On inversion in spanish and some of its effects. *Linguistic Inquiry*, 15, 1984.

[48] Pim van der Eijk and Ton van der Wouden. A modular lexicon environment for NLP. In U. Zernik, editor, *Proceedings of the First International Lexical Acquisition Workshop*, 1989.

[49] Gertjan van Noord. BUG: A directed bottom-up generator for unification based formalisms. *Working Papers in Natural Language Processing, Katholieke Universiteit Leuven, Stichting Taaltechnologie Utrecht*, 4, 1989.

[50] Gertjan van Noord. An overview of head-driven bottom-up generation. In Robert Dale, Chris Mellish, and Michael Zock, editors, *Current Research in Natural Language Generation*. Academic Press, 1990.

[51] Gertjan van Noord. Reversible unification-based machine translation. In *Proceedings of the 13th International Conference on Computational Linguistics*, 1990. to appear.

[52] Gertjan van Noord, Joke Dorrepaal, Doug Arnold, Steven Krauwer, Louisa Sadler, and Louis des Tombe. An approach to sentence-level anaphora in machine translation. In *Fourth Conference of the European Chapter of the Association for Computational Linguistics*, 1989.

[53] Henk van Riemsdijk. *A Case Study in Syntactic Markedness: The binding nature of prepositional phrases.* Foris Publications, Dordrecht, 1978.

[54] Jürgen Wedekind. Generation as structure driven derivation. In *Proceedings of the 12th International Conference on Computational Linguistics*, 1988.

[55] Henk Zeevat, Ewan Klein, and Jo Calder. Unification categorial grammar. In Nicholas Haddock, Ewan Klein, and Glyn Morrill, editors, *Categorial Grammar, Unification Grammar and Parsing.* Centre for Cognitive Science, 1987. Volume 1 of Working Papers in Cognitive Science.