An Overview of MiMo2 *

Gertjan van Noord
Joke Dorrepaal
Pim van der Eijk
Maria Florenza
Herbert Ruessink
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 ideas of Landsbergen [22] 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 [22]). 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.

---

*MiMo2 is a sideline project of the Dutch Eurotra team in Utrecht. An extended version of this article was presented at the MT workshop in Austin, june 1990.

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 [9]) 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 [23, 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 [22].

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. [18, 19]). It may even be the case that one cannot know whether the meaning expressed in two languages is the same (cf. [30]). 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-Eskimo translation of 'snow' in a multilingual system that has Dutch as one of its languages (as Dutch has several different words for 'snow' [29, 24]).

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 [37, 27, 35]. Some recent developments are reported in [10, 46, 36, 42, 8, 38, 39, 43, 40].

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. Reversible systems are preferable to nonreversible ones. The arguments in favour of using bidirectional grammars in NLP, such as those given in [2, 17] carry over to translation. Furthermore Isabelle [16] 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 [35] for a general introduction), Head-Driven Phrase Structure Grammar (HPSG, [28]) 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, [47]), 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 [28, p 7]. As opposed to linguistic theories (such as transformational grammar) and NLP formalisms (such as the Eurotra's E-framework [4]) 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 [6]. MiMo2 follows HPSG [28, 33] 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 [35], or the 'aliases' of [31]) to implement a lexical inheritance hierarchy ([11], [28] chapter 8). Furthermore, maintenance of large lexicons is facilitated by a separate lexical preprocessor, which is discussed elsewhere [41].

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 [12]). 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 following lexical entry 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):

```
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⟩
```

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

A unification grammar defined in grammar formalisms such as DCG [27] and PATR [37] usually defines a relation between a string of words and some representation, sometimes called *logical form*. In sign-based approaches such as UCG [47] and HPSG [28], 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. It is also possible to use unification grammars to define other 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 of these grammars is *reversible*; the system consisting of three of such grammars applied in series is reversible too [44].

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 [21, 5, 32]. 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 [7]. Morphological analysis is based on a separate reversible unification grammar in which derivational processes and compounding can be defined [3]. 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. [1].

The unification grammars defined by the user in a PATR like style are compiled into Prolog (an extension to the compilation described in [14]), 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 [26, 25]. Both techniques are optimized by using a set of 'restricted' features [34]. The generator of Mimo2 is a member of the generation family described in [42, 38, 43, 39]. Transfer is implemented as a top-down backtrack search procedure.

3 Illustration

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. [20]), 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. As an example we will give the analysis of (vp-)negation in the following subsection. Some other phenomena are treated in [45]. The second subsection presents some transfer exampels.

3.1 Negation

The most interesting type of negation - from a translational point of view - is vp-negation, so we will leave other types outside this discussion. Neutral vp-negation for English seems quite straightforward as the negation element 'not' appears immediately after the first auxiliary verb. For Dutch, matters are more complex. According to [15], the negated element in principle immediately precedes the verb if the VP does not contain any other scope bearing element at the left of the verb. It appears directly left of the leftmost scope bearing element otherwise. It even attaches to it morphologically in the case of the indefinite article. Relevant data are the following:

dat Jan Marie de foto niet geeft
(that John Mary the picture not gives)
that John doesn't give Mary the picture

dat Jan Marie geen foto geeft
(that John Mary no picture gives)
that John doesn't give Mary a picture

dat Jan Marie niet alle foto's geeft
(that John Mary not all pictures gives)
that John doesn't give Mary all pictures

Our implementation of this phenomenon makes use of 'threading' [26]. The features $\langle$`negin`$\rangle$ and $\langle$`negout`$\rangle$ are used to pass the information through the left corners of the constituents. A `vp` containing a negative marker will thus have the feature $\langle$`negin`$\rangle$ `= negative`, which value will be preserved in translation. The $\langle$`negout`$\rangle$-value of vp's is constrained to be `nonnegative`.
The grammar rule that introduces the negative element is the following (simplified) :

$x_1 \rightarrow$ *niet* x_2
$\langle x_1$ `bar`$\rangle$ `= 0`
$\langle x_2$ `bar`$\rangle$ `= 0`
$\langle x_1$ `head`$\rangle$ `=` $\langle x_2$ `head`$\rangle$
$\langle x_1$ `negin`$\rangle$ `= negative`
$\langle x_1$ `negout`$\rangle$ `= nonnegative`
$\langle x_2$ `negin`$\rangle$ `=` $\langle x_1$ `negin`$\rangle$
$\langle x_2$ `negout`$\rangle$ `= negative`
$\langle x_2$ `neg`$\rangle$ `= negable`

All lexical items carry the information $\langle$`negin`$\rangle$ `=` $\langle$`negout`$\rangle$. Those that may occur as the left-corner of constituents which do not bear scope - e.g. definite determiners in definite NPs - are marked $\langle$`neg`$\rangle$ `= nonnegable`. This prevents the negative element to occur in front of non-scope bearing elements. The indefinite, inherently negated determiner 'geen' ('no') is specified as $\langle$`negin`$\rangle$ `= negative,` $\langle$`negout`$\rangle$ `= nonnegative`. It prevents this determiner from being doubly negated, that is, the rule above does not apply. And besides, it returns the in-value negative which, in the end, will be preserved in translation. Some sample lexical entries and their behaviour in negative vp's illustrate this further:

de : $\langle$`negin`$\rangle$ = $\langle$`negout`$\rangle$, $\langle$`neg`$\rangle$ = `nonnegable` (the)
alle : $\langle$`negin`$\rangle$ = $\langle$`negout`$\rangle$, $\langle$`neg`$\rangle$ = `negable` (all)
geen : $\langle$`negin`$\rangle$ = `negative`, $\langle$`negout`$\rangle$ = `nonnegative` (no)
geven : $\langle$`negin`$\rangle$ = $\langle$`negout`$\rangle$, $\langle$`neg`$\rangle$ = `negable` (give)[1]

...dat Jan Marie de foto niet geeft
.*.dat Jan Marie niet de foto geeft

.*.dat Jan Marie alle foto's niet geeft
...dat Jan Marie niet alle foto's geeft

...dat Jan Marie geen foto's geeft

The threading technique proved to be very useful for English as well. In some special cases, the negation marker can also be realized in more than one way. Compare the following pairs:

The minister promised nothing
The minister didn't promise anything

The minister never makes promises
The minister doesn't ever make promises

The only difference between 'anywhere' and 'nowhere' (and similarly 'anything' and 'nothing') is that the first has an in-value equal to the out-value whereas the latter has the in-value set to `negative` and the out-value to `nonnegative`. Translation of the Dutch structure a) will yield either b) or c) in English :

a) De minister belooft niets
b) The minister doesn't promise anything
c) The minister promises nothing

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

[1] This holds for verbs in general.

input to transfer may be a feature structure such as:

$$\begin{bmatrix} gb = \begin{bmatrix} pred = open\_fire\_on \\ arg1 = \begin{bmatrix} pred = soldier \\ number = pl \end{bmatrix} \\ arg2 = \begin{bmatrix} pred = \begin{bmatrix} pred = dutch \end{bmatrix} \\ arg1 = \begin{bmatrix} pred = \begin{bmatrix} pred = prime \end{bmatrix} \\ arg1 = \begin{bmatrix} pred = minister \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ neg = neg \end{bmatrix} \\ nl = \end{bmatrix}$$

for a sentence such as 'The soldiers have not opened fire on the Dutch prime minister' where *gb* contains the English logical form and *nl* the Dutch logical form [2].

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 *nl* 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 *nl* will be considered the output of transfer. An example of a simple rule in PATR notation is given as:

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

The integers 0–3 are names of feature structures, where 0 is used to represent the mother node and $1 \ldots n$ represent the daughter nodes. Application of this rule to the feature structure results in the following instantiation, where framed integers represent reentrancies. Transfer then continues by instantiating the feature structures 1, 2 and 3. The Dutch logical form is thus gradually instantiated as

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

transfer proceeds by decomposing the English logical form:

$$\left[\begin{array}{ll}
0 = & \left[\begin{array}{ll} gb = \ldots & \\ nl = & \left[\begin{array}{l} pred = \boxed{1} \\ arg1 = \boxed{2} \\ arg2 = \boxed{3} \\ neg = neg \end{array}\right] \end{array}\right] \\
1 = & \left[\begin{array}{l} gb = open_fire_on \\ nl = \boxed{1} \end{array}\right] \\
2 = & \left[\begin{array}{ll} gb = & \left[\begin{array}{l} pred = soldier \\ number = pl \end{array}\right] \\ nl = \boxed{2} & \end{array}\right] \\
3 = & \left[\begin{array}{ll} gb = & \left[\begin{array}{ll} pred = & \left[\ pred = dutch\ \right] \\ arg1 = & \left[\begin{array}{l} pred = \left[\ pred = prime\ \right] \\ arg1 = \left[\ pred = minister\ \right] \end{array}\right] \end{array}\right] \\ nl = \boxed{3} & \end{array}\right]
\end{array}\right]$$

An example of the rule for the first daughter will be a 'lexical entry' and may look as in:

```
0 →
⟨0 gb⟩ = open_fire_on
⟨0 nl⟩ = het_vuur_openen_op
```

The complex English expression 'prime minister' has to be translated as a simple expression in Dutch: 'premier'. This rule can be defined as:

```
0 →
⟨0 nl pred⟩ = premier
⟨0 gb pred pred⟩ = prime
⟨0 gb arg1 pred⟩ = minister
⟨0 gb arg1 number⟩ = ⟨0 nl number⟩
```

where it is assumed that the construction is analysed in English as an ordinary noun-adjective construction, and where the logical form of the adjective takes the logical form of the noun as its argument. Note that this example of complex transfer is similar to the famous 'schimmel - white horse' case. As a result of the rule applications the value of the *nl* attribute will get instantiated resulting in:

$$\begin{bmatrix} gb = \ldots \\ nl = \begin{bmatrix} pred = het\_vuur\_openen\_op \\ arg1 = \begin{bmatrix} pred = militair \\ number = pl \end{bmatrix} \\ arg2 = \begin{bmatrix} pred = \begin{bmatrix} pred = nederlands \end{bmatrix} \\ arg1 = \begin{bmatrix} pred = premier \end{bmatrix} \end{bmatrix} \\ neg = neg \end{bmatrix} \end{bmatrix}$$

from which the generator generates the string 'De militairen hebben het vuur niet geopend op de Nederlandse premier'.

Further examples and some discussion can be found in [44, 45].

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

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

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

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

[5] 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*, Copenhagen, 1987.

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

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

[8] 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.

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

[10] 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.

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

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

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

[14] 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.

[15] Heleen Hoekstra. Negation. Master's thesis, Rijksuniversiteit Utrecht, 1988.

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

[17] Paul S. Jacobs. Achieving bidirectionality. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING)*, Budapest, 1988.

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

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

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

[21] 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.

[22] Jan Landsbergen. Isomorphic grammars and their use in the Rosetta translation system. In Margaret King, editor, *Machine Translation, the State of the Art.* Edinburgh University Press, 1987.

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

[24] Laura Martin. Eskimo words for snow: a case study in the genesis and decay of an anthropological example. *American Antrhopologist*, 88(2), 1986.

[25] 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.

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

[27] 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 [13].

[28] Carl Pollard and Ivan Sag. *Information Based Syntax and Semantics, Volume 1*. Center for the Study of Language and Information Stanford, 1987.

[29] G.K. Pullum. The great eskimo vocabulary hoax. *Natural Language and Linguistic Theory*, 7(2), 1989.

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

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

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

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

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

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

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

[37] 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.

[38] 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*, Vancouver, 1989.

[39] Stuart M. Shieber, Gertjan van Noord, Robert C. Moore, and Fernando C.N. Pereira. Semantic-head-driven generation. *Computational Linguistics*, 16(1), 1990.

[40] 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.

[41] 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*, Detroit, 1989.

[42] 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.

[43] 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.

[44] Gertjan van Noord. Reversible unification-based machine translation. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING)*, Helsinki, 1990.

[45] Gertjan van Noord, Joke Dorrepaal, Pim van der Eijk, Maria Florenza, and Louis des Tombe. The MiMo2 research system. In *Proceedings of the Third International Conference on Theoretical and Methodological issues in Machine Translation of Natural Languages*, University of Texas at Austin, 1990.

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

[47] 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, University of Edinburgh, 1987. Volume 1 of Working Papers in Cognitive Science.