Shake-and-Bake Machine Translation

John L. BEAVEN *

Computer Laboratory
University of Cambridge
New Museums Site
Pembroke Street
Cambridge CB2 3QG
UK
E-mail: John.Beaven@cl.cam.ac.uk

Abstract

A novel approach to Machine Translation (MT), called **Shake-and-Bake**, is presented, which exploits recent advances in Computational Linguistics in terms of the increased spread of lexicalist unification-based grammar theories. It is argued that it overcomes some difficulties encountered by transfer and interlingual methods.

It offers a greater modularity of the monolingual components, which can be written with independence of each other, using purely monolingual considerations. These are put into correspondence by means of a bilingual lexicon.

The **Shake-and-Bake** approach for MT consists of parsing the Source Language in any usual way, then looking up the words in the bilingual lexicon, and finally generating from the set of translations of these words, but allowing the Target Language grammar to instantiate the relative word ordering, taking advantage of the fact that the parse produces lexical and phrasal signs which are highly constrained (specifically in the semantics). The main algorithm presented for generation is a variation on the well-known CKY one used for parsing.

A toy bidirectional MT system was written to translate between Spanish and English, and some of the entries are shown.

1 Motivation

The research reported here was motivated by the desire to exploit recent trends in Computational Linguistics, such as the appearance of lexicalist unification-based grammar formalisms for the purposes of machine translation, in an attempt to overcome what are perceived to be some of the major shortcomings of transfer and interlingual approaches.

With a transfer-based MT system, the transfer component is very much language-pair specific, and must be written bearing very closely in mind both monolingual components in order to ensure compatibility. Depending on how much work is done by the analysis and generation components, the tasks carried out by the transfer element may vary, but in general this module is very idiosyncratic and will involve several hundred transfer rules. Writing these transfer rules is the most time-consuming aspect of the design of a transfer-based system, as it must be consistent with both monolingual grammars. The process is therefore error-prone, and the result is not very portable, since the consequences of making changes to the monolingual components may be far-reaching as far as the transfer rules are concerned.

One of the main difficulties with interlingual approaches is what Landsbergen [Landsbergen 87] refers to as the *subset problem*. If the system is to be robust, it is essential to guarantee that any interlingual formula derived from any Source Language (SL) expression is amenable to generation into the Target Language (TL). If the interlingua is powerful enough to represent all the meanings in all the languages involved, there will be several (probably infinitely many) formulae in that interlingua which are logically equivalent to the one produced by the analyser. It cannot then be guaranteed that this formula comes under the coverage or the TL generator, unless we can draw logical inferences in the interlingua. The complexity of this task may be computationally daunting, since sub-problems of this (such as satisfiability and non-tautology) are known to be NP-complete ([Garey and Johnson 1979]).

The approach presented here bears some similarity with that of [Alshawi et al 91], which uses

*The work reported here was carried out at the University of Edinburgh under the support of a studentship from the Science and Engineering Research Council. Thanks to Ann Copestake, Mark Hepple, Antonio Sanfilippo, Arturo Trujillo, Pete Whitelock and the anonymous reviewers for their comments. Any errors remain my own.

the algorithm of [Shieber et al. 90] for generation from quasi-logical forms. On the other hand, generation here takes place from a set of TL lexical items, with instantiated semantics, which makes the task easier.

This approach was tested with independently-written grammars for small yet linguistically interesting fragments of Spanish and English, which are used both for parsing and generation. These are put into correspondence by means of a bilingual lexicon containing the kind of information one might expect to find in an ordinary bilingual dictionary.

2 The grammar formalism

A version of Unification Categorial Grammar (UCG) ([Calder et al. 88]) is used. Like many other current grammatical formalisms ([Shieber 86], [Pollard and Sag 87], [Uszkoreit 86]), it represents linguistic objects by sets of feature (or attribute)-value pairs, called *signs*. The values of these signs may be atomic, variables or further sets feature-value pairs. They can therefore be represented as directed acyclic graphs or as attribute-value matrices using the PATR-II notation of [Shieber 86]. The notion of *unification* is then used to combine these.

The main features used in the signs are ORTHOGRAPHY, CAT (the categorial grammar syntax), ORDER (the directionality of the "slash", which specifies linear ordering), FEATS (a set of syntactic features), CASES (a case-assignment mechanism built on top of standard UCG), and SEM, a unification-based semantics with a neo-Davidsonian treatment of roles ([Parsons 80, Dowty 89]). The semantics of an expression is of the form *I:P*, where *I* is a variable for the semantic index of the whole expression, and *P* is a conjunction of propositions in which that index appears. In addition, features called ARG0, ARG1 and so on provide useful "handles" for allowing the bilingual lexicon to access the semantic indices, but they are not strictly necessary for the grammars

The signs presented are only shorthand abbreviations of the full ones used, and the interested reader is referred to [Beaven 92] for a more complete view. The PATR-II notation will be used, with the Prolog convention that names starting with upper case stand for variables. In addition, for the sake of clarity and brevity, the non-essential features will be omitted, as will be their names when these are are obvious.

The grammar rules used subsume both functional application and composition, but for the examples given here, only functional application will be necessary.

An important feature of this approach is that this will make it possible to have an MT system in which no meaningful elements in the translation relation are introduced syncategorematically (in the form of transfer rules or operations with interlingual representations). In particular, assuming we have very rich lexical entries (which contain information about various dimensions of the language, such as orthography, syntax and semantics), all that is needed is a correspondence between the lexical entries, supplied by a bilingual lexicon, together with a set of constraints for each correspondence.

The design of such a translation system will therefore involve three components: two monolingual lexicons for the languages concerned, and a bilingual lexicon. The Spanish and English components were designed using purely monolingual considerations, and as a consequences the treatments of English and Spanish grammars are quite different.

The basics of the grammar will be explained by presenting the monolingual lexical entries required for the Spanish sentence *María visitó Madrid*, which corresponds to the English *Mary visited Madrid*. More linguistically interesting sentences will be offered at a later stage.

2.1 The Spanish Grammar

The Spanish grammar is somewhat an unconventional version of UCG, in that VPs are treated as sentences (S), and NPs as sentence modifiers (S/S in the categorial notation). The reasons for this decision have to do with accounting for subject pro-drop, and are discussed in [Whitelock 88] and [Beaven 92]. A case-assignment mechanism is added to standard UCG. Amongst other uses, it provides a coverage of clitic placement.

NPs are sentence modifiers. The following one, for instance, looks for a sentence with semantics *I1: Sem1*, and returns another sentence, in which the semantics have been modified to state that *F3* (an index standing for *María*), plays a certain (unspecified) role in the semantics of *I1*. The operation ∪ stands for set union, and all the propositions in the semantics are interpreted here as being conjoined.

$$(1)\quad \begin{bmatrix} \text{ORTHO} & \text{'María'} \\ \text{CAT} & s/\begin{bmatrix} s \\ \text{I1} : \text{Sem1} \end{bmatrix} \\ \text{SEM} & \text{I1} : (\left\{\begin{matrix}\text{role(I1,\_R1,F3),} \\ \text{name(F3,maria)}\end{matrix}\right\} \cup \text{Sem1}) \\ \text{ARG0} & \text{F3} \end{bmatrix}$$

Since intransitive verbs are sentences, a transitive verb must be a sentence looking for its object NP (now S/S), which makes sure that this object gets identified with index *Y* (which fills the patient role). This is carried out by the case-assignment mechanism, not shown here. The following entry for the transitive verb will be derived from the base form and abstract tense morphemes (see below).

$$(2)\quad \begin{bmatrix} \mathrm{ORTHO} & \text{visitó} \\ \mathrm{CAT} & s/\begin{bmatrix} s/\begin{bmatrix} s \\ E:\left\{\begin{array}{l}\text{visitar(E)},\\ \text{role(E,agt,X)},\\ \text{role(E,pat,Y)}\end{array}\right\}\end{bmatrix} \\ \text{Sem}\end{bmatrix} \\ \mathrm{SEM} & \text{Sem} \\ \mathrm{ARG0} & E \\ \mathrm{ARG1} & X \\ \mathrm{ARG2} & Y \end{bmatrix}$$

The third NP used just parallels the first one:

$$(3)\quad \begin{bmatrix} \mathrm{ORTHO} & \text{'Madrid'} \\ \mathrm{CAT} & s/\begin{bmatrix} s \\ \text{I3 : Sem3}\end{bmatrix} \\ \mathrm{SEM} & \text{I3} : (\left\{\begin{array}{l}\text{role(I3,\_R3,L3)},\\ \text{name(L3,madrid)}\end{array}\right\} \cup \text{Sem3}) \\ \mathrm{ARG0} & \text{L3} \end{bmatrix}$$

Signs (2) and (3) combine by means of function application to produce the following sentence:

$$(4)\quad \begin{bmatrix} \mathrm{ORTHO} & \text{visitó Madrid} \\ \mathrm{CAT} & s \\ \mathrm{SEM} & E:\left\{\begin{array}{l}\text{visitar(E)},\\ \text{role(E,agt,X1)},\\ \text{role(E,pat,L3)},\\ \text{name(L3,madrid)}\end{array}\right\} \\ \mathrm{ARG0} & E \\ \mathrm{ARG1} & \text{X3} \\ \mathrm{ARG2} & \text{L3} \end{bmatrix}$$

It does not subcategorize for anything, but it may be modified by the NP (3) to give the following sentence:

$$(5)\quad \begin{bmatrix} \mathrm{ORTHO} & \text{María visitó Madrid} \\ \mathrm{CAT} & s \\ \mathrm{SEM} & E:\left\{\begin{array}{l}\text{visitar(E)},\\ \text{role(E,agt,F3)},\\ \text{name(F3,maria)},\\ \text{role(E,pat,L3)},\\ \text{name(L3,madrid)}\end{array}\right\} \\ \mathrm{ARG0} & E \\ \mathrm{ARG1} & \text{F3} \\ \mathrm{ARG2} & \text{L3} \end{bmatrix}$$

Since Spanish word order is relatively free (and in particular since the OVS ordering is possible), the verb does not put tight constraints on the directionality of the NPs. The case-assignment mechanism, which identifies the indices of the NPs, can be used to interact with the ORDER feature if this is desired. In the above example, the only thing that prevents the assignment of agent role to *Madrid* and patient role to *María* are constraints on the semantic types of the arguments of the verb.

2.2 The English Grammar

The English grammar is virtually taken "off the shelf" and closely resembles that of [Calder et al. 88], with only the addition of a case-assignment mechanism (not shown here). A simple NP is as follows:

$$(6)\quad \begin{bmatrix} \mathrm{ORTHO} & \text{'Mary'} \\ \mathrm{CAT} & np \\ \mathrm{SEM} & \text{G3} : \{\text{name(G3,mary)}\} \\ \mathrm{ARG0} & \text{G3} \end{bmatrix}$$

A transitive verb subcategorizes for its object and its subject NPs. Again, the following one is derived from that of the base form and abstract inflectional morphemes:

$$(7)\quad \begin{bmatrix} \mathrm{ORTHO} & \text{visited} \\ \mathrm{CAT} & s\ /\begin{bmatrix} np \\ \text{X2:Sem4}\end{bmatrix} / \begin{bmatrix} np \\ \text{Y2:Sem5}\end{bmatrix} \\ \mathrm{SEM} & \text{E2:} \left(\begin{array}{l}\left\{\begin{array}{l}\text{visiting(E2)},\\ \text{role(E2,agt,X2)},\\ \text{role(E2,pat,Y2)}\end{array}\right\}\\ \cup\ \text{Sem1} \cup \text{Sem2}\end{array}\right) \\ \mathrm{ARG0} & \text{E2} \\ \mathrm{ARG1} & \text{X2} \\ \mathrm{ARG2} & \text{Y2} \end{bmatrix}$$

The remaining NP is:

$$(8)\quad \begin{bmatrix} \text{ORTHO} & \text{'Madrid'} \\ \text{CAT} & \text{np} \\ \text{SEM} & \text{F3} : \{\text{name(F3,mary)}\} \\ \text{ARG0} & \text{F3} \end{bmatrix}$$

2.3 Structure of the bilingual lexicon

The bilingual lexicon merely puts into correspondence pairs of monolingual lexical entries. In other words, each entry in the bilingual lexicon will contain a pair of pointers to monolingual entries in each of the languages translated. These monolingual entries are very rich signs, and the bilingual entries may add constraints for their monolingual signs to be in the translation relation. For instance, if a word has more than one translation depending on how various semantic features become instantiated, the bilingual lexical entries may express these restrictions.

The bilingual lexicon writer needs to be aware of what the monolingual lexicons look like, in order to encode the restrictions that the bilingual sign imposes on the monolingual entries. As long as some broad conventions are followed, this task becomes very straightforward. Most bilingual correspondences are very simple, and merely require some semantic indices in the monolingual signs to be unified. Provided these indices are made easily available in predictable places of the monolingual signs, the task of writing the corresponding lexical entries is very simple. When some semantic constraints need to be put on these indices, again it is a straightforward task. It is only on the occasions when syntactic constraints have to be included that the monolingual signs need to be examined more closely, in order to determine how that syntactic information is encoded.

This results in a great modularity in the system. Any monolingual component may easily be changed, without affecting to any significant extent the bilingual lexicon, and certainly not the monolingual components for any other language. At the same time, the simplicity of the bilingual component makes it practicable to write multi-language systems, since all the hard work goes into the monolingual lexicons which may be reused for many language pairs, and the language-pair-specific information is concisely kept in the bilingual lexicon.

The following examples represent entries in the bilingual lexicon. Such an entry consists of pointers to monolingual signs (for instance, (9) puts signs (1) and (6) into correspondence), together with constraints about the semantic indices contained in these signs. Thus example (9) identifies the semantic indices of the two monolingual signs.

$$(9)\quad \begin{bmatrix} \text{SPANISH} & \boxed{1} \begin{bmatrix} \text{SEM} & \begin{bmatrix} \text{ARG0} & \text{F3} \end{bmatrix} \end{bmatrix} \\ \text{ENGLISH} & \boxed{6} \begin{bmatrix} \text{SEM} & \begin{bmatrix} \text{ARG0} & \text{F3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$(10)\quad \begin{bmatrix} \text{SPANISH} & \boxed{2} \begin{bmatrix} \text{ARG0} & \text{E} \\ \text{ARG1} & \text{X} \\ \text{ARG2} & \text{Y} \end{bmatrix} \\ \text{ENGLISH} & \boxed{7} \begin{bmatrix} \text{ARG0} & \text{E} \\ \text{ARG1} & \text{X} \\ \text{ARG2} & \text{Y} \end{bmatrix} \end{bmatrix}$$

(The above is not exactly the entry as it appears in the bilingual lexicon, since correspondences between morphemes are used, but it clarifies the exposition).

$$(11)\quad \begin{bmatrix} \text{SPANISH} & \boxed{3} \begin{bmatrix} \text{SEM} & \begin{bmatrix} \text{ARG0} & \text{F3} \end{bmatrix} \end{bmatrix} \\ \text{ENGLISH} & \boxed{8} \begin{bmatrix} \text{SEM} & \begin{bmatrix} \text{ARG0} & \text{F3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In this very simple example, there was a one-to-one correspondence between monolingual entries. More generally, the bilingual lexicon will encode correspondences between *sets* of monolingual entries, with appropriate constraints on them (which allows us to enter idioms in the bilingual lexicon). Most of the time these will be singletons, but they may occasionally contain several elements or indeed one of them may be empty (if a word in one language corresponds to the empty string in the other, as will sometimes occur with function words).

3 Shake-and-Bake

A new algorithm for generation, developed by Pete Whitelock and Mike Reape, and known as **Shake-and-Bake** is presented (see [Whitelock 92] for further discussion). It can be outlined as follows: first of all the SL expression is parsed using the SL (monolingual) grammar. After the parse is complete the variables in the semantic indices are Skolemised, and lexical entries are looked up in the bilingual lexicon and replaced with their TL equivalents. Generation then takes place starting from the bag of TL lexical entries, which have their semantic indices instantiated as a result of the parsing and look-up process.

Two well-known parsing algorithms (shift-reduce and CKY) have been adapted to do this kind of generation instead. Generation in this context can be seen as a variation of parsing, in which we let the syntactic constraints instantiate the word order rather than letting the word order drive the parsing process.

The CKY parsing algorithm may be characterised as follows: it uses a chart or table where all well-formed substrings (WFSs) that are found are recorded, together with their position (i.e. the words that they span in the string). The table is initialised with the n words of the input string. The algorithm builds parses by finding the shorter WFSs before the longer ones. For all integers j between 2 and n, it records all WFSs of length j by looking for two adjacent strings of length k and $j-k$ recorded on the table. If they may combine by means of a grammar rule, the result is recorded on the table.

The algorithm may be modified for generating strings from a bag of lexical entries. The table here no longer records the position of WFSs, but just the WFSs with the set of entries from the bag that they are made from. It is initialised by recording first all the well-formed strings of length 1 (the lexical entries). Then, for all integers j from 2 to n (the cardinality of the bag), it looks for two disjoint WFSs of length k and $j-k$ recorded in the table. If they combine by means of an (unordered) grammar rule, the resulting string (with orthography specified by the direction of the combination) is recorded on the table, together with the set of entries it involves (the union of the sets of the two components).

Starting from the bag of TL signs above, this algorithm would first put the verb and object together into a component, and then combine the result of that with the subject of the sentence. Linear ordering is determined by the TL grammar and the fact that the semantic indices are instantiated by the time generation takes place.

4 Morphology and Further examples

Finally we shall see how Shake-and-Bake handles more interesting examples, in particular those involving argument switching and head switching.

Entries for verbs such as the ones shown above are derived from the base forms and single morphemes. For instance, *visited* is derived from morphemes for *visit*, *3sg* and *past*. A similar thing is done for Spanish, and the bilingual lexicon actually puts into correspondence the base forms and the separate morphemes. Correspondences between morphemes will be used from here on.

4.1 Argument switching

Argument switching, such as *John likes Mary*, which translates into Spanish as *María gusta a Juan* (literally *Mary pleases John* can be covered in a very simple manner. The monolingual verbs closely resemble (2) and (7).

Their essential features are just:

(12)
$$\begin{bmatrix} \text{ORTHO} & \text{like} \\ \text{SEM} & \text{E1:}\left\{\begin{array}{l}\text{like(E1)},\\ \text{role(E1,experiencer,X1)},\\ \text{role(E1,stimulus,Y1)}\end{array}\right\} \\ \text{ARG0} & \text{E1} \\ \text{ARG1} & \text{X1} \\ \text{ARG2} & \text{Y1} \end{bmatrix}$$

(13)
$$\begin{bmatrix} \text{ORTHO} & \text{gust-} \\ \text{SEM} & \text{E2:}\left\{\begin{array}{l}\text{gustar(E2)},\\ \text{role(E2,stimulus,X2)},\\ \text{role(E2,experiencer,Y2)}\end{array}\right\} \\ \text{ARG0} & \text{E2} \\ \text{ARG1} & \text{X2} \\ \text{ARG2} & \text{Y2} \end{bmatrix}$$

The bilingual entry merely needs to cross-identify the semantic indices:

(14)
$$\begin{bmatrix} \text{SPANISH} & \boxed{(13)}\begin{bmatrix}\text{SEM} & \begin{bmatrix}\text{ARG0} & \text{E}\\ \text{ARG1} & \text{X}\\ \text{ARG2} & \text{Y}\end{bmatrix}\end{bmatrix} \\ \text{ENGLISH} & \boxed{(12)}\begin{bmatrix}\text{SEM} & \begin{bmatrix}\text{ARG0} & \text{E}\\ \text{ARG1} & \text{Y}\\ \text{ARG2} & \text{X}\end{bmatrix}\end{bmatrix} \end{bmatrix}$$

4.2 Head switching

A harder example is when the head word in one language corresponds to a non-head in the other, such as *Mary swam across the river*, which translates as *María cruzó el río nadando* (literally *Mary crossed the river swimming*).

This can be solved by putting into correspondence *across* with the stem *cruz-* as a possible translation pair, together with the base form *swim* with *nadando*. The morphemes for *3sg* and *past* are also put into correspondence.

$$(15)\quad \begin{bmatrix} \text{ORTHO} & \text{across} \\ \text{CAT} & s/\left\{\begin{matrix} \begin{bmatrix} \text{CAT} & \text{np} \\ \text{SEM} & \text{Crossed:Sem1} \end{bmatrix} \\ \begin{bmatrix} \text{CAT} & s \\ \text{SEM} & \text{E1:Sem2} \\ \text{ARG0} & \text{E1} \\ \text{ARG1} & \text{Crosser} \end{bmatrix} \end{matrix}\right\} \\ \text{SEM} & \text{E1} : \left(\begin{matrix} \text{Sem1} \cup \text{Sem2} \cup \\ \{\text{role(E1,across,Crossed)}\} \end{matrix} \right) \\ \text{ARG0} & \text{E1} \\ \text{ARG1} & \text{Crosser} \\ \text{ARG2} & \text{Crossed} \end{bmatrix}$$

$$(16)\quad \begin{bmatrix} \text{ORTHO} & \text{cruz-} \\ \text{CAT} & s/\text{NP} \\ \text{SEM} & \text{E2} : \left\{ \begin{matrix} \text{cruzar(E2),} \\ \text{role(E2,agt,Crosser),} \\ \text{role(E2,pat,Crossed)} \end{matrix} \right\} \end{bmatrix}$$

The bilingual entry that puts these two together is:

$$(17)\quad \begin{bmatrix} \text{SPANISH} & \boxed{(15)} \begin{bmatrix} \text{ARG0} & \text{E} \\ \text{ARG1} & \text{Crosser} \\ \text{ARG2} & \text{Crossed} \end{bmatrix} \\ \text{ENGLISH} & \boxed{(16)} \begin{bmatrix} \text{ARG0} & \text{E} \\ \text{ARG1} & \text{Crosser} \\ \text{ARG2} & \text{Crossed} \end{bmatrix} \end{bmatrix}$$

A similar pair of monolingual entries, together with the bilingual entry to put them into correspondence, is needed for *swim-nadando.*

$$(18)\quad \begin{bmatrix} \text{ORTHO} & \text{swim} \\ \text{CAT} & s/\begin{bmatrix} \text{CAT} & \text{NP:X} \\ \text{SEM} & \boxed{a}\ \text{E3:}\left\{ \begin{matrix} \text{swim(E3),} \\ \text{role(E3,agt,X)} \end{matrix} \right\} \end{bmatrix} \\ \text{SEM} & \boxed{a} \\ \text{ARG0} & \text{E3} \\ \text{ARG1} & \text{X} \end{bmatrix}$$

$$(19)\quad \begin{bmatrix} \text{ORTHO} & \text{nadando} \\ \text{CAT} & s/\begin{bmatrix} \text{CAT} & \text{S} \\ \text{SEM} & \text{E4:Sem} \\ \text{ARG0} & \text{E4} \\ \text{ARG1} & \text{X} \end{bmatrix} \\ \text{SEM} & \text{E4:}(\{\text{nadar(E4)}\} \cup \text{Sem}) \\ \text{ARG0} & \text{E4} \\ \text{ARG1} & \text{X} \end{bmatrix}$$

$$(20)\quad \begin{bmatrix} \text{SPANISH} & \boxed{(18)} \begin{bmatrix} \text{ARG0} & \text{E} \\ \text{ARG1} & \text{X} \end{bmatrix} \\ \text{ENGLISH} & \boxed{(19)} \begin{bmatrix} \text{ARG0} & \text{E} \\ \text{ARG1} & \text{X} \end{bmatrix} \end{bmatrix}$$

The important aspects of these signs is that the bilingual element correctly identifies the indices of the lexical entries, and the Shake-and-Bake generation takes care of the rest.

5 Conclusion

I hope to have shown how lexically-driven Machine Translation makes it possible to write modern, unification-based monolingual grammars with great independence from each other, and to put them into correspondence by means of a bilingual lexicon of a similar degree of complexity as one might expect to find in a commonly available bilingual dictionary, which could make it easier to automate its construction.

These points were demonstrated by constructing two monolingual Unification Categorial Grammars for small fragments of Spanish and English, which nevertheless included some linguistically interesting phenomena. They were written independently, and with purely monolingual considerations in mind, which led to some noticeable differences in the grammar design. The monolingual components were put into correspondence by means of a bilingual lexicon, and algorithms for parsing, doing bilingual lookup and generation were suggested, which together constitute what has been named **Shake-and-Bake Translation.**

While the process of Shake and Bake generation itself is NP-complete, it is likely that average case complexity may be reasonable ([Brew 92]). In this sense, Shake and Bake may address issues raised by the Landsbergen's subset problem, since inference in an interlingua may not even be decidable.

References

[Alshawi et al. 91] Alshawi, H., Carter, D., Rayner, M., and Gambäck, B. Translation by Quasi Logical Form Transfer. In *Proceedings of the 29th Annual Meeting of the Association of Computational Linguistics*, pages 161-168, Berkeley, 1991.

[Beaven 92] Beaven, J.L. *Lexicalist Unification-Based Machine Translation*, PhD Thesis, University of Edinburgh, 1992.

[Brew 92] Brew, C. Letting the cat out of the bag: generation for Shake-and-Bake MT. *Proceedings of the 14th International Conference on Computational Linguistics (COLING 92)*, Nantes, 1992.

[Calder et al. 88] Calder, J., Klein, E. and Zeevat, H. Unification Categorial Grammar — A Concise, Extendable Grammar for Natural Language Processing. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING 88)*, pages 83–86, Budapest, 1988.

[Dowty 89] Dowty, D. On the Semantic Content of Notion "Thematic Role". In Chierchia, G., Partee, B. and Turner, R. (eds.) *Property Theory, Type Theory and Natural Language Semantics.* Dordrecht: D. Reidel, 1989.

[Garey and Johnson 79] Garey, M.J., and Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman & Co, New York, 1979.

[Landsbergen 87] Landsbergen, J. Montague Grammar and Machine Translation. In Whitelock, P. J., Wood, M. M., Somers, H., Bennett, P., and Johnson, R. (eds.) *Linguistic Theory and Computer Applications.* Academic Press, 1987.

[Parsons 80] Parsons, T. Modifiers and Quantifiers in Natural Language. *Canadian Journal of Philosophy*, supplementary Volume VI, pages 29-60, 1980.

[Pollard and Sag 87] Pollard, C. and Sag, I.A. *Information-Based Syntax and Semantics - Volume 1: Fundamentals.* Lecture Notes Number 13. Center for the Study of Language and Information, Stanford University, 1987.

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

[Shieber et al. 90] Shieber, S., van Noord, G, Pereira, F.C.N, and Moore, R.C. Semantic-Head-Driven Generation. *Computational Linguistics*, Volume 16, number 1, pages 30-42, 1990.

[Uszkoreit 86] Uszkoreit, H. Categorial Unification Grammars. *Proceedings of the 11th International Conference on Computational Linguistics (COLING 86)*, pages 187–194, Bonn, 1986.

[Whitelock 88] Whitelock, P. *A Feature-based Categorial Morpho-Syntax for Japanese.* DAI research paper no 324, Dept. of Artificial Intelligence, Univ. of Edinburgh. Also in Rohre, C., and Reyle, U. (eds.) *Natural Language Parsing and Linguistic Theories.* D. Reidel, Dordrecht, 1988.

[Whitelock 92] Whitelock, P. Shake and Bake Translation. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING 92)*, Nantes, 1992.