

REVERSIBLE MACHINE TRANSLATION: WHAT TO DO WHEN THE LANGUAGES DON'T LINE UP

James Barnett, Inderjeet Mani, Paul Martin, and Elaine Rich

MCC

3500 West Balcones Center Dr.

Austin, TX 78759

Abstract

In this paper we deal with issues that face an interlingua-based, reversible machine translation system when the literal meaning of the source text is not identical to the literal meaning of the *natural* target translation. We present an algorithm for lexical choice that handles such cases and that relies exclusively on reversible, monolingual linguistic descriptions and a language-independent domain knowledge base.

1 Introduction

Machine translation is an obvious application for reversible natural language systems, since both understanding and generation are important parts of the process. There are several arguments for this view (for example, [Isabelle, 89]), including reducing the total cost of adding a new language and making it easier to maintain and validate the resulting system.

Reversible MT systems, just like the broader class of MT systems as a whole, fall into two roughly defined families: transfer systems and interlingua (or pivot systems). Reversible transfer systems (e.g., [van Noord, 90], [Zajac, 90], [Dymetman, 88], and [Strzalkowski, 90]) exploit three reversible subsystems: one to analyze the source text, one to perform the transfer, and a third to generate the target text. Interlingua-based systems (e.g., Ultra [Farwell, 90]), on the other hand, require only two reversible components: one to analyze the source text into the interlingua representation, and one to generate the target text from that representation. In this paper, we will focus on issues that arise in the design of interlingua-based MT systems.

The simplest model of a reversible, interlingua-based system contains two components: one analyzes the source text to create the interlingua representation and the other maps from that to the target text. Unfortunately, the real situation is

not that simple, for several reasons, including two that we will focus on here:

- This model assumes that the same information is present in the target text as in the source. But in some cases, which have been called *translation mismatches* [Kameyama, 91], information is either added to or deleted from the source in creating the target. We will show some examples of this below in Section 2. In these cases, the simple reversible system we outlined above would produce unacceptable translations.
- Although the notion of a reversible system that describes the set of legal translations is reasonably clearcut, the notion of *preferred translation* is more difficult to define [van Noord, 90], [Barnett, 91d]. In some cases, which have been called *translation divergences* [Dorr, 90], the most natural translation differs from the source in some significant way (e.g., its focus).

Of course, in many cases, both of these issues occur together and interact. In this paper, we present some techniques for dealing with these problems. These techniques have three important properties: They require purely declarative, reversible descriptions of the languages that are involved. They require only monolingual facts. Thus new languages can be added to the system without any changes to the descriptions of any other languages. And they are stated in a way that enables their performance to increase gradually along with the power of the underlying knowledge base.

2 Translation Divergences and Mismatches

In this section, we examine some examples in which the source and target languages do not line

up. Then, in the rest of the paper, we will outline our solution to these problems.

1. **English:** “The *dogs* were running down the street.”

Japanese: “*inu ga toori-o hashitte-ita.*” (lit. “dog run (along) the street.”)

In English, noun phrases must be marked for number. In the natural Japanese translation, number information is absent.

2. **English:** “I saw a *fish* in the water.”

Spanish: “Vi un *pez* en el agua.”

English: “I ate a *fish*.”

Spanish: “Comi un *pescado*.”

Spanish makes a distinction between a fish in its natural state (“pez”) and a fish that has been caught for food (“pescado”). “Pez” is also the default form in case it is not clear or does not matter what state the fish is in. But it cannot be used if it is clear that the fish has been caught. To get the translation right, it is necessary to infer extra information about the fish, using other knowledge that is available either from the rest of the sentence or from the larger discourse context. Similarly, to reverse the process and go from Spanish to English, it is necessary, in the case of “pescado”, to throw away information lest we produce the unnatural translation, “I ate a caught fish.” It is important to note, though, that this information cannot be thrown away during understanding, since it would be important if we were translating into another language that made the same distinction. It must be preserved until the point at which generation into the target language takes place.

3. **English:** “I *know* him.”

Spanish: “Lo *Conozco*.”

English: “I know the answer.”

Spanish: “*Se* la respuesta.”

Here the issue is the correct translation between English “know” and the two Spanish verbs “conocer” (to be acquainted with someone) and “saber” (to know a fact). This example is similar to the previous one except that here there is no default form. Spanish does not have a word that includes these two different events.

4. **English:** “I have a headache.”

Japanese: “Atama ga itai.” (literally, “my head hurts”)

Here the problem is more difficult. No longer is it an issue of a single lexical item for which there is not an exact match in the target language. Instead, the texts in the two languages differ at the level of an entire phrase, with each language choosing a phrase that describes the situation from a different point of view. In English, we seem to describe an *object*, “a headache”, while Japanese describes the *state* of a head hurting.

The examples that we have just discussed illustrate three different categories of semantic differences between languages:

- Mismatches caused by semantically significant differences in morphology and syntax, e.g., Example 1. Other common examples involve the presence or absence of markings for gender, number, tense, aspect, and level of politeness.
- Mismatches caused by lexical differences, where one language has a word that the other lacks, e.g., Examples 2 and 3.
- Divergences, in which the two languages describe the same state of the world in different ways, as in Example 4. In some of these cases, identical information is conveyed (in the sense that the semantic interpretation of the source implies that of the target and vice versa), but in some cases (and depending on the particular model of the world that is being used to define implication) the semantic content of the two forms will not be identical, so many cases of divergence also contain mismatches.

Mismatches and divergences are typically viewed as translation (transfer) problems. But in an interlingua-based system it becomes clear that they are primarily problems for generation. The source language analyzer produces an interlingua representation, which the target generator must render into the target language. In cases of mismatch or divergence, doing this requires manipulating the interlingua expression itself since it does not already correspond exactly to the structure of the target string that should be produced. But actually, the fact that the expressions in the

interlingua representation came from linguistic expressions in a source language as opposed to from some other source (for example, the output of a problem-solving system) is irrelevant except for a few special cases in which the form of the source language expressions can provide help in making generation decisions. So, in the rest of this paper, we will present a generation-centered treatment of mismatches that relies entirely on reversible, monolingual descriptions of the two languages.

3 The KBNL MT System

Figure 1 shows a schematic description of the MT system that we are building. All of the representations in the figure, except the source and target language strings, are described in terms that are drawn from a knowledge base (KB) that describes the domain(s) of discourse. In addition to providing a common set of terms that enable meanings to be defined, this backend knowledge base is important because it provides the ability to reason about meanings and thus the ability to add to the target text information that was omitted from the source. We will assume that all the KB-based representations can be treated as sets of logical assertions (although they can of course be implemented in a variety of ways, including the frame-based system [Crawford, 90] that we are using).

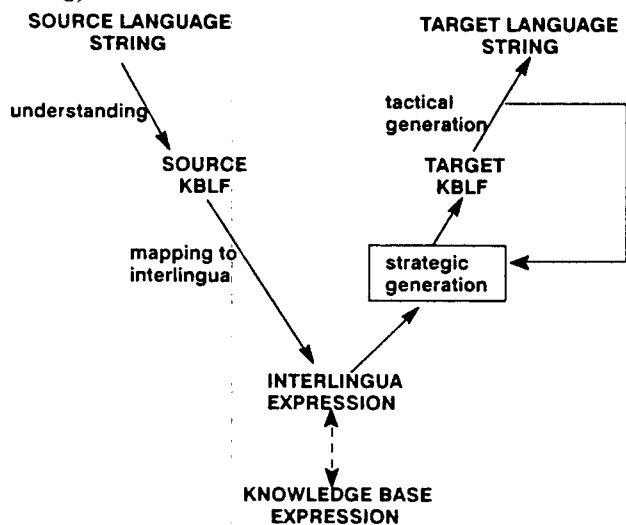


Figure 1: An Interlingua-Based Architecture for MT

To translate a sentence, this system must do the following things:

- Map the source sentence into an internal representation of what was said. We call this

the *source DRS*; it is isomorphic to the Discourse Representation Structures described in [Kamp, 84] and [Heim, 82], except that its terms are taken from the backend knowledge base rather than from the words of the source language.

- Map the source DRS into the interlingua, which is equivalent to the source DRS, both in form and in content. Thus it contains assertions corresponding to exactly what was said in the source.
- Map the interlingua expression to a target DRS. At this point, decisions about what to say in the target text must be made. Some assertions in the interlingua may be dropped. Some new assertions may be added. Some groups of assertions may be replaced by others that are equivalent with respect to the KB but more appropriate as a basis for a natural sounding text in the target language.
- Map the target DRS into a target string. Unfortunately, it is often not possible to enforce a clean separation between these last two generation steps, so it may be necessary for them to interact and to inform each other, as shown in by the loop in the figure.

We have implemented an MT system for English and Spanish in this framework. It is based on the KBNL system [Barnett, 91a], which has two key components: Lucy a language understanding system, and Koko, a language generation system. Both Lucy and Koko use a common agenda-based blackboard for communication and control. And they both exploit a generic KB interface [Barnett, 91b], so they can run on any KB that contains the necessary domain knowledge. We assume (in contrast to some other interlingua-based MT systems, e.g., [Uchida, 89]), that the KB, and thus the interlingua, has not been designed with any particular set of languages in mind.

Lucy and Koko have been designed to use a single, reversible linguistic description [Barnett, 90], so that a language need only be specified once and can then serve as both a source and a target. The syntactic component of this system is based on an extension of Categorical Unification Grammar, which serves as the phrase-structure component of an LFG-style f-structure representation. Semantic processing in both systems is mostly compositional, and is driven by a shared lexicon that describes the meanings of words in

terms of the backend KB. Declarative rules for handling phenomena such as metonymy and noun compounding are also shared between the two systems, although they are compiled into separate forms to support understanding and generation. We have used this approach to build a reversible English/Spanish MT system.

Since much of the discussion below will center around strategies for lexical choice during generation, we will devote the rest of this section to a brief description of Koko's generation algorithm. In the current implementation, Koko handles only the tactical generation phase of Figure 1. It takes as its input a DRS that contains the meaning that is to be realized, and, optionally, an f-structure that describes the syntactic form that the realization should take¹. In Section 6, we will discuss extending it to handle the task of generating the best target DRS. In addition to a set of semantic assertions, the DRS contains a distinguished variable that points to the discourse entity that the source utterance is 'about'. For example, in Example 2 above, this discourse entity would be the fish.

Given this input as a goal, Koko uses the semantic-head driven algorithm described in [Calder, 89] to generate a phrase whose syntax and semantics satisfy the goal (this algorithm is a special case, suited for categorial grammars, of the algorithm described in [Shieber, 90]). The algorithm works by peeling off lexical functors and recursing on their arguments until it bottoms out in an atomic constituent². At each recursive step of this algorithm, a lexical look-up procedure is invoked. This procedure attempts to find a lexical item that matches the current goal. Once this lexical item, called the semantic head, is found, the algorithm proceeds both top-down and bottom up. If the semantic head is a functor, it proceeds top-down trying to solve the sub-goal(s) for its argument(s). We use here a notion of goal satisfaction where a solution (a constituent) *sat-*

¹The f-structure can be as specific as desired. It may contain no more than the target category or it may even specify which words to use. We do not make use of f-structure specifications in the lexical choice algorithms discussed here.

²In a categorial grammar, most of the syntactic information is contained in the lexical items. For example, where a phrase-structure grammar might have a rule $S \rightarrow NP V NP$, a categorial grammar will assign the category $S \setminus NP / NP$ to a verb. The category says, in effect, that the verb wants to combine with an NP to its right, and one to its left to form a full S. Any such constituent that takes at least one argument is called a *functor*, while a constituent with no arguments is called *atomic*.

isfies a goal if it has identical semantics and its f-structure is a supergraph of the goal's f-structure. Once a sub-goal is satisfied, the algorithm works bottom-up by applying (unary) grammar rules to the argument constituent alone, or (binary) rules to combine it with the functor. The algorithm terminates when a complete constituent that satisfies the goal is found.

We now describe the lexical choice component of this generation procedure in more detail. This component is driven by a reverse index that organizes words by the KB concepts that occur in the word's meaning. To find a lexical item that satisfies a particular generation goal, the lexical choice procedure performs a kind of classification operation; it looks at the semantic assertions in the goal and finds candidate words that match some or all of those assertions. Words that operate syntactically as functors are acceptable even if they match only partially; the recursive part of the process will attempt to match the remaining assertions with words that can serve as the functor's arguments. Words that operate syntactically as atomic constituents must match all the assertions in order to succeed since there is no additional way to match any assertions that are left over.

Unfortunately, in the simple form in which it was just stated, this algorithm for lexical choice fails to handle cases of semantic mismatch between source and target languages. This is because it takes as input the assertions that were derived from the source text and expects to generate a target text that *exactly* covers those same assertions. In the rest of this paper, we describe modifications to this algorithm that handle cases such as the ones in Examples 1-4.

4 Forced/Unforced Distinctions

Semantic mismatches of the kind shown in Example 1 arise from morphological differences between languages. When an inflection is syntactically obligatory in a language and it also carries semantic information, a speaker of that language is forced to specify facts that can be left out in other languages. For example, speakers of English are forced to specify number on NPs, which Japanese does not require. Speakers of Japanese, in turn, have to indicate the level of formality of the discourse as well as the social relation between the participants. Verb tense, on the other hand,

is obligatory in both languages.

To implement this, we alter the grammar of each language to mark as *Forced* all assertions that come from syntactically obligatory inflections. The marking indicates that the assertion is forced, and records the type of inflection (e.g., number or tense) that forced it. Then we must consider two modifications to the basic procedure for lexical look-up: one in which forced assertions from the source text can be dropped from the target because they are not required and one in which there are forced distinctions in the target and the corresponding assertions were not present anywhere in the source (i.e., they are not forced in the source nor was the information explicitly volunteered.)

We first consider the case in which forced assertions from the source are not also required in the target language. In general they should be dropped. The exception is when there is an assertion that carries important information and would have been volunteered but did not have to be since it was forced anyway. This is relatively rare, detecting it is in general difficult, and it requires reasoning within the current discourse context. We describe here what happens if we assume that the forced assertion should not be carried over. To handle this, we modify the procedure for lexical look-up to accept partial matches in which assertions that are marked as having been forced in the source language but that are not forced by the target grammar are ignored. In Example 1, for instance, we will allow “inu”, which has no number assertions, to match the goal³

(dog x) (> (quantity x) 1)

Notice, though, that we will still reject any proposed match that conflicts with a forced assertion.⁴ For example, if there is a forced singular assertion in the source we will not allow a plural lexical form to be used in the target. But we will accept as a match a word that makes no commitment at all about number.

The more difficult case is the one in which the target language forces a distinction that is not made in the source. In this case, some information must be added to the target text. In some cases, the information can be derived from the larger discourse context. In other cases, it may be possible to ask the user. And, if both of these

³We are using a representation of plurals and mass terms based on [Link, 83].

⁴Two assertions conflict if they assign incompatible values to a slot/property of an object.

fail, the system must have a default. This case is very similar to what happens in Examples 2 and 3, in which the lexicons of the source and target languages fail to match. In all three cases, there is no target word that corresponds exactly to the set of source assertions but there are some number of target words that correspond to the source assertions augmented with some additional information. We will deal with this problem in detail in the next section.⁵

5 Lexical Choice

Now we consider those cases in which differences in the lexicons of the source and target languages cause assertions to be either added or dropped.

To solve this problem, we need to introduce the notion of marked and unmarked lexical forms.⁶ We define this notion as follows. Consider a set *S* of objects or events (which may or may not be a class), and assume that the lexical item *L* is associated with *S*. Now consider one or more specializations (subsets) of *S*, each of which is defined to have some particular value along some relevant dimension. The case we are concerned with is the following:

1. There is some subset *SS* along some dimension *D* and there is a lexical item *LL* (distinct from *L*) associated with *SS*. In other words, there is a specialized word for this specialized class.
2. Although *L* can be used to describe any element of *S* whose value along dimension *D* is unknown, it is infelicitous to use *L* rather than *LL* to describe an object that is clearly an element of *SS*. By “clearly” here we mean by inspection of the nearby context of the discourse.

In this case, we define *L* to be an unmarked form along dimension *D* and *LL* to be a marked form.

To illustrate this definition, we return to the pez/pescado example. Let *S* be the set of fish. In Spanish, *L* is then “pez”. But there is a subset *SS* of caught fish, and *LL* is “pescado”. It is infelicitous to use the word “pez” when it is clear from context that the fish has been caught. So

⁵See, in particular, step 5a for a treatment of exactly this case.

⁶The marked/unmarked distinction that we are exploiting here is analogous to the more traditional one that is used in morphology [Jakobson, 66].

“pez” is unmarked along the dimension of being caught, and “pescado” is marked. The English word, “fish”, is neither marked nor unmarked.

It is important to note here that the choice between a general word and more specific words does not always involve a distinction between marked and unmarked terms. For example, the choice (in English or Spanish) between “fish” and words for its subclasses “trout”, “salmon”, etc. is *free* in the sense that it is perfectly acceptable to use “fish” even when we know the object in question is a trout (unless the fact that it is a trout is relevant to the conversation, in which case we are violating Gricean principles.)

Though there seem to be some cross-linguistic generalities about markedness (e.g., that markedness is rare along dimensions that are defined by natural classes), it is a language-specific fact that certain words are marked along certain dimensions, and these facts must be acquired along with the grammar of the language. Acquiring these distinctions will be a substantial amount of work, but the work is necessary even in non-reversible monolingual systems. For example, a Spanish language question answering system needs to know that the choice between “trucha” (trout) and “pez” is different (and freer) than that between “pez” and “pescado”, and that “pez” is the default for the latter distinction. Thus, the use of markedness in our lexical choice algorithm is independently motivated, and is not something that has to be added just to get reversible machine translation to work.

We can now state the algorithm for lexical choice. This algorithm appears to be a complex enumeration of a set of special cases, and in some sense it is. The reason is that it is actually two processes overlaid on top of each other. The first is a generation process that deals with the need to add and subtract information but that does not depend on the fact that the DRS it is working with came from a linguistic source. The second is the fact that there are a few places where facts about the source text and the source lexicon can be used to provide guidance to the general purpose generation algorithm. For a longer discussion of the interaction between these two processes, see [Barnett, 91c].

The lexical choice procedure takes as input a list of assertions that describe a set *S* of objects or events. The list is structured, with all assertions arising from a single source lexical item grouped together.

There are places in this algorithm where appeal is made to a knowledge base, its associated infer-

ence mechanisms, and a knowledge-based model of the current discourse context. We mark these places with (§). The performance of this algorithm is tied to the ability of the underlying KB to provide accurate answers to these questions either by reasoning or by asking a user. In each case, we describe a default strategy that can be used in the case of incomplete knowledge in the KB.

There are also places in the algorithm where considerations of meaning alone allow more than one possible lexical choice, and stylistic factors must be considered. We mark these places with (#). The performance of this algorithm in these cases is tied to our ability to extract statements of style from the source text and to use those statements, as well as stylistic preferences within the target language, to make choices that best achieve the desired style.

Algorithm: Modified Lexical Choice

1. If there is a word for *S* in the target language, then we want to do a straightforward translation except in the case where there was also a single word for *S* in the source language but the speaker chose not to use it and to use a descriptive phrase instead (for example, in a definition of the word).⁷ In that case, we need to preserve that free choice by using a phrase in the target as well. So check to see if there is a single word for *S* in the source but the assertions that define *S* came from more than one lexical item. In this case, split the assertions into two subgoals, one for the head and one for the modifiers and recursively call this algorithm.
2. If there is a word *W* for *S* in the target language and the redundancy check defined above failed, then if *W* is not unmarked in the target lexicon, use it. If there is more than one, then (#) choose the one with the style that best matches the style of the source.
3. If there is a word *W* for *S* in the target language but it is unmarked along some set of dimensions *D*, then we need to see if we should use one of the more specific marked forms rather than *W*. (For example, “fish” in English will map to the unmarked Spanish form

⁷Notice that checking for this case would not be necessary in a straightforward transfer system. It is only an issue here because we want to be able, when appropriate, to use words that are available in the target but were not in the source.

“pez”.) So for each element of D, examine all of the available marked forms. For each of them, do:

- (a) Check to see if there is a corresponding marked word in the source language. If there is, then since it was not used in the source we do not need to consider using it in the target either, so we can skip this form.
- (b) Otherwise, (\$) check (using some fixed effort level) to see whether the additional information that would license this form can be inferred from the discourse context. If it can, then select that form. (For example, the information that licenses “pescado” will be available for the source sentence, “I ate fish for dinner.”) If there are synonymous marked forms, (#) use style as a basis for choosing.

If none of the marked forms is chosen, then use the unmarked form.⁸

4. There is no word for S in the target language. (For example, this happens in translating Spanish “pescado” or “pescado blanco” into English, or English “know” into Spanish.) In this case, we must do one of two things:

- See if there is a more specific word that can be shown to be applicable.
- Use a more general word and add modifiers as necessary to communicate the additional information.⁹

Neither of these operations can be done on an entire phrase at once. So we must peel assertions off and pass them and the remainder of the assertion list recursively to this algorithm. But we need to distinguish between additional information that was volunteered (e.g., “blanco”) and so should definitely be rendered in the target, and additional information that was forced by the lexicon of the source language (e.g., the fact that the fish had been caught). So we need to keep together all the assertions that came from a

⁸It could in principle happen, if there are lots of dimensions, that more than one marked form will be found. We have not found any examples of this, though, so we have not considered how to choose among them.

⁹See [Sondheimer, 88] for a discussion of various possibilities in picking additional modifiers.

single source word. To do that, we must peel off groups of assertions that came from single lexical items rather than individual assertions.

An additional complication is that there may be a single word in the target language for a combination of modifiers that required several words in the source. Or there may be a word for the head combined with a modifier other than the last one. The only way to find such words is to peel off modifiers in all possible orders one at a time, two at a time, three at a time, and so forth. So, if the assertions that describe S came from more than one lexical item in the source, examine all combinations of ways to peel off modifiers (keeping together assertions that came from a single lexical item), and recursively invoke this algorithm on the peeled off part and the remainder, doing the remainder first and stripping from the peeled off part any assertions that are subsumed by the choice of a rendering for the remainder. If more than one distinct target expression results from this process, (#) use the target language stylistic rules to choose among them.

5. There is no word for S in the target language and all the assertions that describe S came from a single lexical item in the source. (For example, this happens in translating Spanish “pescado” into English or English “know” into Spanish or Japanese “inu” into English.)

(a) First consider the possibility that there is a word that is more specific in the sense that it supplies morphological information that is required (forced) in the target language. If there is a set of such words, call that set SS.

- i. For each element of SS, (\$) check to see whether the additional information that would license it can be inferred from the discourse context (just as in Step 3b above). If it can, then select that word.
- ii. If there is not enough information present in the context to license any of the elements of SS, then select the one that is labeled as default.

This path will handle the case we described in Section 4 where a syntactic distinction that was absent in the source text is forced in the target language.

For example, it will handle translating Japanese “inu” into English: since the concept Dog will point to both the singular and plural forms of “dog”, one of these forms must be chosen.

(b) If there was no set SS in the last step, we next consider the possibility that there is a word that is more specific in some other way. Loop until there are no further specializations of S for which the target language contains lexical items:

- i. Let SS be the set of immediate specializations of S (the first time through) or the previous value of SS minus all rejected entries (all other times).
- ii. For each element of SS, check to see whether it or any of its specializations is lexicalized in the target language. If not, eliminate it (and all its descendants) from further consideration.
- iii. For each remaining element of SS, (\$) check to see whether the additional information that would license it can be inferred from the discourse context (just as in Steps 3b and 5a above). If it can, and if it itself is lexicalized, select its lexicalization. (For example, in translating English “know” into Spanish, this step should succeed for either “saber” or “conocer”.)

If, during step iii, the additional requirements for any element of SS are proven to be unsatisfiable in the current discourse context, eliminate it (and its descendants) from further consideration.

(c) If no more specific word is found, we must use a more general one. (\$) Trace up the knowledge base generalization hierarchy from S until a set that does have a rendering in the target language is found. (For example, in translating “pescado”, we trace up to the concept Fish.) Call this P and recursively invoke this algorithm to realize P in the target language. If there is more than one candidate for P, then follow all paths for the remainder of this algorithm and (#) use stylistic rules, such as brevity or preservation of focus, to choose among

the resulting expressions. This particular path will result in the translation of Spanish “pescado” as “fish”.

(d) We must also compute the set of assertions that would enable a classifier to distinguish S from P (in other words, all the information that we would be throwing away if we just described S as P). Call this C. Now we need to decide whether to translate C. We should do that if C was volunteered in the source but not if it was forced by the source lexicon. So check the source lexicon for P. If there is an entry that is not unmarked on any dimension included in C, then the additional information was volunteered. Recursively invoke this algorithm on C to render it. If there is no entry or there is one that is unmarked on one or more dimensions included in C (as it will be in the case of the concept Fish that we will use in translating “pescado” in Example 2) then do:

- i. For each such dimension, (\$) check (using a fixed effort level) whether the information given is both nonobvious (i.e., it will not be inferable by the reader of the target from context) and important for the sense of the text. If it can be shown to be,¹⁰ then recursively invoke this algorithm to render it. Otherwise (as for example with the fact that the fish was caught), drop it.
- ii. For all the remaining assertions in C, recursively invoke this algorithm.

6 Translation Divergence

Now we briefly consider cases of translation divergence, such as the one in Example 4 above. There must be two parts to the solution to this problem. First we consider the case in which, for a given DRS, there is more than one grammatically

¹⁰ As an example of a case where it is necessary to render such information, consider translating the Japanese word “gohan” into English. “Gohan” is the unmarked form for rice. It also means specifically “cooked rice”, in contrast to the marked form, “kome”, which means raw rice. Suppose that “gohan” is being used in a recipe that specifically requires cooked rice. Then it is important that the modifier “cooked” be rendered explicitly because it matters, yet it is not inferable since raw rice is also a possible (and in fact even more common) ingredient.

acceptable rendering, but one is preferred. Here, it is necessary to extend the notion of markedness so that it applies not just to individual lexical items but also to grammatical structures. Just as in the lexical case, a marked form, if it is applicable, must block the use of any unmarked form. The natural forms must then be marked, and they will block the use of “grammatical” but unnatural forms. One common way to implement this notion of a marked grammatical form is to use phrasal lexicons in which the preferred forms are listed directly and the more general grammar is only used when no stored phrases match.

But we must also consider the case in which the natural form cannot be generated directly from the DRS. Rather, it is first necessary to derive a related (possibly equivalent) DRS and then to generate from that. This is the process that we described as strategic generation in Figure 1. But now the question arises: how do we choose among the candidate DRSs and their corresponding target strings? The answer is again that marked forms should block unmarked ones. The simplest way to implement this is to derive all the equivalent DRS structures, generate from all of them, and then rank the results. There may be more efficient ways of doing this, particularly in the case that patterns of marked forms can be used to compile preferences into DRS forms, but we have not yet begun to look seriously at this issue.

7 Conclusion

In this paper, we have described an approach to machine translation that has three important properties:

- It treats many problems of translation mismatch and divergence as primarily problems of generation from a flexible semantic representation language rather than as translation problems *per se*.
- It relies exclusively on reversible, monolingual descriptions of all of the languages it treats. Although some comparisons of the source and target lexicons are required, they can be done automatically (and cached if desired). No language-pair information must be explicitly provided.
- It is stated in a way that enables its performance to increase steadily with the performance of the underlying knowledge base and reasoning system.

This approach does, however, require some additional information that is not normally present either in monolingual NL systems or MT systems. Some of this information must be provided as part of the definition of each language. This includes:

- The labeling of syntactic assertions as forced or unforced. This information is only useful for MT, but it is very easy to provide.
- The labeling of marked/unmarked distinctions along various dimensions. This requires more work, but it is also useful even in purely monolingual generation systems, since they may be given sets of assertions for which there is no exact match.

Some additional information must also be passed along during the understanding process. In particular, the grouping together of assertions that came from the same lexical item must be preserved.

8 Acknowledgements

We’d like to thank the other members of the KBNL project: Chinatsu Aone, Jim Blevins, Bill Bohrer, Dilip D’Souza, Susann Luper-Foy, Kevin Knight, Juan Carlos Martinez, and David Newman for their contributions to this paper.

References

- [Barnett, 90] J. Barnett and I. Mani, “Using Bidirectional Semantic Rules for Generation”, *Proceedings of the Fifth International Workshop on Natural Language Generation*, pp.47-53, Dawson, Pa., 3-6 June, 1990.
- [Barnett, 91a] J. Barnett, D. D’Souza, K. Knight, I. Mani, P. Martin, E. Rich, C. Aone, J. Blevins, W. Bohrer, S. Luper-Foy, J.C. Martinez, and D. Newman, “Knowledge-Based Natural Language Processing: the KBNL System”, MCC Technical Report ACT-NL-123-91, 1991.
- [Barnett, 91b] J. Barnett, E. Rich, and D. Wroblewski, “A Functional Interface to a Knowledge Base for Use by a Natural Language Processing System”, MCC Technical Report ACT-NL-019-91, 1991.

- [Barnett, 91c] J. Barnett, I. Mani, E. Rich, C. Aone, K. Knight, and J. C. Martinez, "Capturing Language-Specific Semantic Distinctions in Interlingua-Based MT", in *Proceedings of MT Summit 3*, Washington, D.C., 1991.
- [Barnett, 91d] J. Barnett and I. Mani, "Shared Preferences", *Proceedings of the ACL Workshop on Reversible Grammars in Natural Language Processing*, Berkeley, 1991.
- [Calder, 89] J. Calder, M. Reape, and H. Zeevat, "An Algorithm for Generation in Unification Categorical Grammar", *Proceedings of the 4th Conference of the European Chapter of the ACL*, pp. 233-240, Manchester, 10-12 April, 1989.
- [Crawford, 90] J. Crawford, "Access-Limited Logic—A Language for Knowledge Representation", Ph.D. Thesis, The University of Texas at Austin, 1990.
- [Dorr, 90] B. Dorr, "Solving Thematic Divergences in Machine Translation", *Proceedings of the 28th Annual Meeting of the ACL*, Pittsburgh, 1990.
- [Dymetman, 88] M. Dymetman and P. Isabelle, "Reversible Logic Grammars for Machine Translation", *Proceedings of the Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, 1988.
- [Farwell, 90] D. Farwell and Y. Wilks, "Ultra: A Multi-Lingual Machine Translator", Memoranda in Computer and Cognitive Science MCCS-90-202, Computing Research Laboratory, New Mexico State University, 1990.
- [Heim, 82] I. Heim, "The Semantics of Definite and Indefinite Noun Phrases", *University of Massachusetts*, Ph.D. Dissertation, 1982.
- [Isabelle, 89] P. Isabelle, "Towards Reversible MT Systems", in *Proceedings of MT Summit II*, 1989.
- [Jakobson, 66] R. Jakobson, "Zur Struktur des Russischen Verbums", in Hamp, Householder, and Austerlitz, eds., *Readings in Linguistics, II*, Chicago: The University of Chicago Press, 1966.
- [Kameyama, 91] M. Kameyama, R. Ochitani, and S. Peters, "Resolving Translation Mismatches with Information Flow", *Proceedings of the 29th Annual Meeting of the ACL*, Berkeley, 1991.
- [Kamp, 84] H. Kamp, "A Theory of Truth and Semantic Representation", in M. Groenendijk, J. Janssen, and M. Stokhoff, eds., *Formal Methods in the Study of Language*, Dordrecht: Foris, 1984.
- [Link, 83] G. Link, "The Logical Analysis of Plurals and Mass Terms: A Lattice-Theoretical Approach", in R. Baeuerle et al. (eds.), *Meaning, Use and Interpretation*. Berlin: de Gruyter, 1983.
- [Shieber, 90] S. Shieber, G. van Noord, F. Pereira and R. Moore, "Semantic-Head-Driven Generation", *Computational Linguistics* 16(1), March, 1990.
- [Sondheimer, 88] N. K. Sondheimer, S. Cumming, and R. N. Albano, "How to Realize a Concept: Lexical Selection and the Conceptual Network in Text Generation", in *Proceedings of the Workshop on Theoretical and Computational Issues in Lexical Semantics*, 1988.
- [Strzalkowski, 90] T. Strzalkowski, "Reversible Logic Grammars for Parsing and Generation", *Computational Intelligence* 6(3), 1990.
- [Uchida, 89] H. Uchida and M. Zhu, "An Interlingua for Multilingual Machine Translation", *Shizengengoshori, 1989. 5. 19.*, Japan, 1989.
- [van Noord, 90] G. Van Noord, "Reversible Unification-Based Machine Translation", in *Proceedings of COLING-90*, Helsinki, 1990.
- [Zajac, 90] R. Zajac, "A Relational Approach to Translation", in *Proceedings of the Third International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, Austin, Texas, 1990.