# Lexical Issues in Dealing with Semantic Mismatches and Divergences

**James Barnett**
**Elaine Rich**

**MCC**

**barnett@mcc.com**
**rich@mcc.com**

In this paper, we address the question, "What types of MT divergences and mismatches must be accommodated in the lexicon?" In our work, we have focused on the treatment of divergences and mismatches in an interlingual (as opposed to a tranfer-based) MT system. In such systems, one uses monolingual lexicons for each of the language under consideration, so divergences and mismatches are not handled as explicitly as they must be in transfer-based systems that rely on bilingual lexicons. In an interlingual system, these issues become issues in NL generation into the target language.

## Divergences and Mismatches

Two kinds of problems can occur as a result of a divergence [Dorr 90] or a mismatch [Kameyama 91]. The first is the relatively simple case, in which there is a lexical gap, i.e., a straightforward attempt to render a concept that was mentioned in the source language fails to find the required word. When this happens it is clear that there is a problem, although it is less clear what to do about it.

Example: English *know* <--> German *wissen/kennen*

The more difficult case is the one in which the lexicon provides a way to express the exact (or almost exact) concept that was extracted from the source text but there is also at least one alternative way to express the same (or similar) thing that is more natural in the target language. In this case, even noticing the problem is not necessarily straightforward. We consider two subcases here:

- The problem is with a single word. This occurs when there are word pairs one of which is a default and the other of which is marked. In these cases, the default may only be used when the marked case cannot be used. For example, in Spanish the word *pez* (fish) is a default -- it can only be used if the more specific word *pescado* (caught fish suitable for food) is not appropriate.[1]

- The problem is at the phrasal level. In these cases, although each word is acceptable in some context, the larger phrase sounds awkward. There are several specific kinds of circumstances in which this occurs, including:
    1. The main event has more than one argument/modifier, one of which can be incorporated, along with the main event, into the main verb, with the others being added as syntactic arguments/modifiers. There may be lexical preferences for choosing one argument over another for incorporation. For example,

        French: Il a traversé la rivière à la nage.
        English: He traversed the river by swimming.
        He swam across the river.

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

In this example, the main event is physical motion. In French, the fact that the motion was all the way to the other side is incorporated into the main verb and the kind of motion (swimming) is added as a modifier. Although this same incorporation is possible in English (as shown in the literal translation), the alternative, in which the kind of motion is incorporated into the main verb and the completion of the motion is indicated with a preposition is more natural.

2. Although there is a general kind of phrase that can be used to express the desired concept, there is also a more specialized idiom (or semi-idiom) that is applicable in the current situation and should therefore be used. For example, in English, you can say, "I have a pain in my ____" for just about any part of your body. But in a few special cases (e.g., *head*) it is more natural to say, "I have a ____ache."

Our goal is to define a tactical generation system that takes as its input a structure we call a discourse-level semantic structure (DLSS). The DLSS is composed of a set of referents that will correspond to the objects to be mentioned and a set of assertions about those referents. The assertions will be in terms defined by the knowledge base that underlies the interlingua. Because the DLSS already corresponds roughly to the way that the target text will be expressed, the techniques we are developing are unable to handle cases where global transformations must be applied to the result of understanding the source text. In those cases, additional mechanisms (probably understood right now by no one) must be applied to the result of the source language understanding process before the tactical generation process into the target language can begin.

We want to produce a target text that has two properties:

- It comes as close as possible to rendering exactly the semantic content of the DLSS that was input to the tactical generator.
- It sounds as natural as possible.

These two dimensions are orthogonal. But they cannot be considered independently. Sometimes, a very small change in semantics may enable a significant change in naturalness. So, the task of designing a generation algorithm within this framework can be divided into the following tasks:

1. Define what it means to be the best rendition, in a given language, of an input DLSS. This definition must include some way of trading off accuracy and naturalness.
2. Define a search procedure that will, in principle, find all the correct renditions of an input DLSS. Any reasonable generation algorithm will work here.
3. Describe a set of heuristics that will make the search procedure tractable.

**Viewing the Problem as One of Generation**

As suggested above, we view this entire process as one of generation, rather than as translation *per se*. But even the brief description of it that has so far been presented makes it clear that there is a sense in which what we are doing is transforming the interlingua expression (which resulted from analyzing the source text) into an expression in the target language. At one level, this is very similar to conventional stuctural transfer of the sort that takes place in most transfer-based MT systems. But there is a crucial difference. There are no transformations that are driven by relationships between linguistic structures in one language and linguistic structures in another. Instead, there are transformations that are driven by the lexicon and grammar of the target language, which can be thought of as forcing the transformations to produce the most natural, semantically acceptable output in the target language. True, the input to the transformations comes from analyzing expressions in another language, and so the form clearly does

depend on the lexicon and the grammar of the source language. And, in fact, we will use some information about the source lexicon in some of our generation heuristics. But in general, the process by which text is generated needs to know little or nothing about the source language.

The only real sense in which the needs of MT have pushed our work on generation in directions different from more generic efforts to build NL generation systems is the following: In some constrained generation environments, it is possible to get away with assuming that the input to the generator (from some external source, such as a problem solving system) is already in a form that is very close to the form that the final linguistic output should take. This is particularly likely to be true if the input to the generator is based on a knowledge base that was designed with one specific language in mind (as is often the case unless the need to support natural representations of expressions in other languages is explicitly considered while the KB is being built). What MT does is to force us to consider cases in which the input to the generator is in a form quite different from the final output form. And, in particular, we must consider cases in which the KB is not an exact match to the lexicon that the generator must use (since it cannot simultaneously correspond exactly to more than one language's lexicon). But the mechanisms for dealing with these cases need not be specific to MT. Instead, they can be applied just as effectively in other generation environments, particularly those in which the underlying KB is sufficiently rich that there can exist forms that do not happen to correspond to natural linguistic expressions in the target language.

In the rest of this paper, we will sketch the answers that we have so far developed to each of the three issues raised above. These answers require that we view the target language lexicon as containing more information than is often provided. In particular, it is necessary not just to include possible words and phrases, but also to indicate for each when it is the most natural way to express the associated concept.

**What is a Correct Rendition of a DLSS?**

To define what it means to be a correct rendition of a given DLSS, we need to do three things:

- Define what we mean by accuracy (the extent to which the target text exactly matches the semantics of the DLSS). We will refer to this as semantic closeness.
- Define what we mean by syntactic naturalness (a function solely of the target text itself).
- Specify a way to handle the tradeoff between accuracy and naturalness

Assume for the moment that we have measures of semantic closeness and naturalness (which we will describe below). One way to handle the tradeoff between the two is to define some kind of combining function. Then a generation algorithm could search the whole space of remotely possible output strings and select the one that was best, given the combining function we chose. There are two problems with this approach. The first is that there appears to be no principled basis for choosing such a function. The second is more pragmatic -- the generation algorithm will waste a lot of time exploring possible output texts that will simple get rejected at evaluation time.

An alternative approach is to take a more structured view of the interaction between the two measures and build them into the algorithm in a way that enables us to constrain more tightly the generation process. We do that as follows.

The generation procedure takes four inputs:

- the DLSS to be rendered into the target language.
- an absolute semantic closeness threshold $\theta$. We will accept as a correct rendition of the

input DLSS only those strings whose semantics is within $\theta$ of the DLSS.

- a semantic closeness interval $\delta(\theta)$.
- a syntactic naturalness threshold N. We will accept as a correct rendition of the input DLSS only those strings whose naturalness is above N.

The generation algorithm them works as follows: Start with a reasonably tight bound on semantic closeness (initially $\delta(\theta)$). See if you can generate one or more acceptably natural sentences within this bound. If so, pick the most natural of the candidates. If not, increase the allowable difference and try again. Continue until you reach the absolute bound on closeness. If there are no acceptably natural sentences within this bound, the algorithm fails.

How tight the original closeness bound is, how loose it is eventually allowed to get, and what level of naturalness are required will be functions of the generation/translation context. In some situations, loose but natural translations will be preferred. In others, tight but perhaps unnatural ones may be better. In any case, by using an iterative approach, it is possible to generate an entire text in which most of the sentences are very close to the input, with only a small number (the ones where the close version sounded really bad) being a bit farther away.

**Semantic Closeness**

How then do we define the semantic closeness measure that this algorithm requires? We want a measure that captures three things:

- How different are the semantics of the DLSS and the target text? For example, if we assume a set theoretic model, how many elements are in one but not the other? We will use the expression $\Delta S$ to refer to the difference between the DLSS and the information content of the proposed text. $\Delta(S)$ is in principle a statement in the semantics language. What we need for the algorithm however is a numeric measure of the size of $\Delta(S)$. We will use $|\Delta(S)|$ to indicate that measure, which will be computed not by actually computing $\Delta(S)$ and then measuring it, but by assigning costs to each of the inference rules that are used to move from one semantic description to another.
- How likely is it that the semantics of the target text acurately characterize the situation that the DLSS is describing? For example, the DLSS may correspond to the meaning of the English word *know* and the target text may end up with semantics corresponding to the German word *kennen*, but if the situation being described is one of being acquainted with a person, then the target text accurately describes that situation even if it is not identical to the English source text. Let P(C) be the probability that the semantics of the target text accurately characterize the situation.
- If there are differences between the semantics of the target text and the situation being described, how much is anyone likely to care about those differences? For example, suppose we translate the English word *vegetable* into the Japanese word *yasai*, which describes a class that mostly but not completely overlaps the class described by *vegetable*. In most texts this difference probably does not matter at all, but in a book on horticulture, it might. Let CARE($\Delta(S)$) be a measure of how much anyone is likely to care about any semantic differences that do exist. In general, if we throw away information, we need to be concerned about whether it was important. If we add information, we need to worry about whether adding it will give the reader some incorrect ideas that matter.

In any real program, it will not be possible to compute any of these measures exactly. But we can exploit heuristics. For the first, we can look at the specific inferences that were made within the knowledge base. For the second, we must also appeal to the KB, although here we may also want to

appeal to the source lexicon as well (see below). For the third, we must appeal again to the KB and also to some model (possibly very simple) of what matters in the domain of the current generation task.

To see how these measures interact to define closeness, let's examine each of the common KB inference procedures that can be used to generate sets of assertions that are likely to be "close" to the input DLSS.

First consider the case in which we move up in the KB hierarchy (in other words we throw out information). This happens in translating the German words *kennen* and *wissen* into English, where we have to move up to the more general concept of knowing. A clearer example occurs in translating the Spanish word *pescado* (meaning a fish that has been caught and is ready to be food) into the English *fish*. In this case, there is no chance of being wrong. So we define the closeness of a proposed text to the input DLSS as simply[2]

$$|\Delta S| * CARE(\Delta(S))$$

Next consider the case in which we move down in the KB hierarchy (in other words, we add information). This happens in translation the English word *fish* into the Spanish *pescado*, or in translating the English *know* into German. Now we have to consider the risk that the information that has been added is wrong. Sometimes the risk is low (for example, because straightforward type constraints on the other components of the sentence guarantee that the required conditions are met.) But sometimes there is a risk. So closeness is defined as

$$|\Delta S| * P(C) * CARE(\Delta(S))$$

Finally, consider the case in which we move sideways, to related situations. This operation is very common, because it occurs in several different kinds of scenarios. It happens in cases (for example English *vegetable* vs. Japanese *yasai*) in which there is no word for the exact concept but there is a word for a very similar one. It also happens in cases of alternative incorporations (as in the *swim/traverse* example). In both cases, it is possible that we are both adding and throwing away information. In the latter case, the likelihood of making a mistake, however, is very low. In the former, it depends on how similar the concepts are. In either case, we use as a measure of closeness,

$$|\Delta S| * P(C) * CARE(\Delta(S))$$

**Syntactic Naturalness**

Now we need to define what it means to be a syntactically acceptable sentence in the target language. It is at this step that the lexicon plays a key role. And it is here that we need to take into account the difference between "sort of in the language" (e.g., "He traversed the river by swimming."), and "natural in the language" (e.g., "He swam across the river.")

There are three possible reasons why something that is "in the language" (i.e., it can be produced using the lexicon and the grammar of the language) might be considered unnatural:

1. A word or phrase that is labeled as a default in a default/marked alternation pair is used when the marked form is appropriate.

[2]We use multiplication in all of these functions. It isn't critical that it be multiplication. Almost any monotonic function will probably work.

2. A phrase that is generally okay is generated but there is a nearly equivalent (semantically) phrase that is preferred. This is what is happening in both the "traverse the river" and the "I have a pain in my head" examples.)
3. There are genuine "negative idioms" in the language. By a negative idiom we mean some phrase that, while generatable from the lexicon and the grammar of the language, is simply not acceptable, for whatever reason. We are not sure yet what the data are on the existence of such "phrasal gaps", but we allow for them in our approach.

Of these, the second is by far the most common and thus the most important from the point of view both of the lexicon itself and of the generation algorithm. Yet this cause of unnaturalness presents a problem for any evaluation function that is intended to look at a proposed output string and evaluate it. To do that, we need to be able to tell, by looking just at the string and at the lexical entries and rules that produced it, how good it is. In the case of default words and negative idioms (cases 1 and 3 above), this is not difficult, because both can be indicated as infelicitous in the lexicon. Default words need just to be marked as such. Negative idioms need only be stored, using the same format as positive ones (see below), along with a rating that indicates the appropropriate level of unacceptability. But this approach doesn't really work in case 2. We do not want to have to enter explicitly into the lexicon every phrase that is "eclipsed" by something better.

Let's look at the "traverse the river" example to make this issue clear. We are trying to translate from French to English. Suppose we start generating with a very small $\delta(\theta)$, because we want a fairly literal translation if we can find a good one. Then we may produce, "He traversed the river by swimming." The lexical entries for these words tell us nothing about the fact that this sounds unnatural, nor does the grammar indicate that any unusual construction was used. So there is no basis for giving this sentence a low acceptability score. Yet we want to, precisely because there is an alternative way to say it that is very, very close semantically and that is a lot better in terms of naturalness.

There are two possible solutions to this problem. The first is to modify the generation algorithm and add a sort of cloud around $\theta$. If anything in the cloud is a lot more natural than the other sentences, then take it. The big drawback to this approach is that it is computationally very expensive. Adding to $\theta$ means firing knowledge base inference rules, and that is likely to be explosive.

The alternative is to run the inference rules in the other direction and cache the results. In other words, start with all the phrases that are marked in the lexicon as being preferred. Take the semantic expressions that correspond to them and, starting at those points, apply the inverses of the inference rules that would normally be used during generation. Any phrases that correspond to the semantic expressions that result should be marked in the lexicon as bad.

In other words, we need to be able to tell when a sentence is bad (i.e., highly unnatural). But we do not want to depend on being given a lot of explcit information about bad things, because there are too many of them and because the badness usually results from the existence of good things (so saying it again would be redundant). Fortunately, however, we can extract the required information automatically. In addition, it is possible to add explicit badness information as a way to tune the lexicon, but this should only be necesssary in cases where the knowledge base is incomplete and it is not worth the effort to fix it.

So we now have two kinds of information that must be present in the lexicon in order for naturalness judgements to be possible. The first is that default/marked pairs must be indicated. This is easy.

The second, and more significant thing is that it must be straightforward to enter phrases, at varying degrees of generality, into the lexicon. Each lexical entry must have the following general form:

```
<conceptual pattern>

<linguistic pattern>

(<style>)

<goodness function>
```

For example, to indicate that we have a preference for using words like swimming (i.e., incorporating the kind of motion into the main verb), we might have the following entry in our English lexicon (using a shorthand for the actual patterns):

```
motion-event (X),
mechanism-of-motion (X,Z)
path X,Y

<word corresponding to Z> <path expression> Y

1
```

Often the goodness function will be a single number. In fact, using simply 0 for ordinary word entries and 1 for phrases that are known to be good in certain circumstances (as described by the stated patterns of their arguments) goes a long way. In some cases, however, it is necessary to use a function, which typically takes stylistic information as its argument. So we might have a technical word whose goodness is 1 if we are attempting to generate technical prose and 0 otherwise. Because stylistic information is not always necessary for computing goodness, it is optional in each lexical entry.

**Choosing the Most Natural Whole Sentence**

So far, we have talked about ways to indicate, in the lexicon, the naturalness of individual words and phrases. But how do we combine those into scores for whole sentences so that we can choose the best?

There are two parts to the answer to this question. The first lets us get a rough measure. The second is then used to compare a set of sentences whose scores are nearly equal given the rough measurement.

To compute the rough measurement of the naturalness of a sentence, just average the scores of the components that were used to compose it. We need to use an average so that the generation algorithm can use a single acceptability threshold regardless of the length of the sentence being generated. By this measure, sentences that contain "bad" words or phrases will have lower scores than those that don't.

But what if there are multiple candidate sentences none of which is "bad"? Now we use a different measure. Here we prefer sentences in which the language does more work to ones where it does less. What we mean is that we prefer:

- The use of inflectional morphology to communicate something over the use of additional lexical items.
- The use of a single word to communicate a set of assertions over a phrase (unless the word is stylistically inappropriate in the current context or there is external information, e.g., the source text, that tells us that the word cannot be used).

- The use of a more restricted phrase to the use of a more general purpose one, and the use of even a general purpose stored phrase to the use of text that is derived a word at a time using the grammar. A phrase A is more restricted than another phrase B if the semantic conditions on the use of A subsume those on the use of B.

These heuristics, taken together, usually but not always prefer shorter sentences to longer ones. For example, they prefer, "He swam across the river," to "He traversed the river by swimming." But "I have a headache," is longer than, "My head hurts," even though it is more idiomatic.

**Using the Source Text for Advice**

So far, we have described the process of target language generation as one in which only the target language dictionary and grammar need be used. There are cases, though, in which the source text and the source lexicon can be exploited for additional information that can be used during generation. We give two examples.

First consider what happens if, in attempting to generate the target text, the initial attempt to find a word for a concept produces a word that is marked as a default (e.g., *pez*). In general, it is necessary to find the corresponding marked word (e.g., *pescado*) and see if its semantic conditions can be proven to be probably true in the current context. If they can, then the marked word must be used. But suppose that the source language has the same default/marked alternation. Then the reason we are looking at the default word is that the corresponding default word was used in the source text. Assuming that text was correctly written, we are guaranteed that the applicability conditions for the marked word are not met. So we do not need to check for them during generation.

As a second example, consider the case in which an entire set of semantic conditions can be accounted for by using a single word, rather than a head word and a set of modifiers. In general, we want to do that. For example, we would want to render the meaning of the English phrase, "nasty little dog," as *roquet* in French. But suppose that the source language also has a single word and didn't use it (perhaps because this occurence of the word is in a definition). Then it is probably a good idea not to use it in the target text either.

Notice that in both these cases, although we make use of the source language lexicon during generation, there is no transfer lexicon. Neither source nor target lexicon needs to know anything about the other.

**Considering More than One Sentence at a Time**

So far, we have considered the problem of generating a single sentence from a single DLSS. We have assumed the existence of a strategic planning component that precedes the actual generation system and whose goal is to construct the appropriate DLSS. In particular, for MT, we assume that often the DLSS can be patterned after the structures in the source text. Sometimes, however, it may be desirable to change some the decisions made by the strategic planner during generation. For example, it might be possible to produce a much more natural sounding sentence if it were allowed to change the focus or the presuppositions of the sentence. If these kinds of transformations are done, their costs can be incorporated into $|\Delta S|$, just as are the costs of other changes to the semantics of the original DLSS. But since text level issues tend to be much more common across languages than lexical and morphological ones are, these kinds of transforms are less important than the ones we have focused on in this

discussion.

**Summary of Implications for the Lexicon**

The main implications of this discussion for lexicons designed to support interlingual MT systems are the following:

- Default/marked pairs must be indicated in the lexicon.
- Phrases that represent prefered ways of saying things need to be entered into the lexicon as phrases even if their semantics can be derived compositionally from the meanings of their constituents (and thus there would be no reason to enter them into a lexicon intended just to support understanding rather than generation).
- Restrictions on the applicability of phrases need to be stated as tightly as possible.
- The lexicon needs to be tied to a knowledge base that supports the kinds of reasoning that we have described for getting from a literal meaning to a related meaning that can be more naturally expressed in the target language.

Notice that in some sense none of these requirements is unique to MT however. All of these things are necessary in any NL generation system that expects to be able to generate from inputs that do not exactly match the way things are typically said in the target language.

## References


[Dorr 90] Dorr, B.
Solving Thematic Divergences in Machine Translation.
In *Proceedings of ACL.* 1990.

[Jakobson 66] Jakobson, R.
Zur Struktur des Russischen Verbums.
In Hamp, Householder, & Austerlitz (editors), *Readings in Linguistics II.* The University of Chicago Press, Chicago, 1966.

[Kameyama 91] Kameyama, M., R. Ochitani & S. Peters.
Resolving Translation Mismatches with Information Flow.
In *Proceedings of the 29th Annual Meeting of the ACL.* 1991.