

EBMT Seen as Case-based Reasoning

Harold Somers

Centre for Computational Linguistics, UMIST,
PO Box 88, Manchester M60 1QD, England
Harold.Somers@umist.ac.uk

Abstract

This paper looks at EBMT from the perspective of the Case-based Reasoning (CBR) paradigm. We attempt to describe the task of machine translation (MT) seen as a potential application of CBR, and attempt to describe MT in standard CBR terms. The aim is to see if other applications of CBR can suggest better ways to approach EBMT.

Introduction

Case-based reasoning (CBR) is a well-established paradigm for problem solving which emerged in the 1980s as an alternative to rule-based expert systems. Instead of rules, CBR represents expertise in the form of past “cases”, and new problems are solved by finding the most similar case in the case-base, and using this as a model for the new solution through a process of “adaptation”.

EBMT is a reasonably well-established paradigm for machine translation which emerged in the 1990s as an alternative to rule-based MT systems. Instead of rules, EBMT represents its linguistic knowledge in the form of “examples” of previous translations, and new translations are made by finding the most similar example in the example-base, and using this as a model for the new translation through a process of “recombination”.

The parallel between CBR and EBMT is so obvious that one would think it perhaps unnecessary to make it. But, despite the earlier establishment of CBR as a problem-solving paradigm, very few papers on EBMT make the connection explicit, and if they do, it is only as a passing comment. With one notable exception, reference to CBR pays only lip service: no attempt is made to take what has been said about CBR to see if it applies to the problem of MT. The major exception, which we should mention very clearly, is the work of Bróna Collins and colleagues (Collins, 1998; Collins & Cunningham, 1995, 1996, 1997; Collins et al., 1996): her work was explicitly in the paradigm of CBR, since it was carried out from within a Computer Science department specialising in CBR. As for the rest of the EBMT literature, the present author has attempted (Somers, 1999) a very thorough survey of articles on EBMT: of about 130 articles collected and read, less than 10% even mentioned CBR or related paradigms, by name.

The purpose of this paper is to look at MT from the perspective of CBR, that is, to consider the CBR approach to problem solving, to see how (or whether) CBR terminology and ideas fit the particular problem of MT, and to see if we can gain any insights from this exercise. The basic assumption of this paper is that EBMT does indeed come within the general paradigm of CBR-based systems. For the purposes of discussion, however, we will use the terms “EBMT” and “CBR” distinctively: the former in its normal meaning, the latter to imply CBR seen as a generic problem-solving method.

CBR: the Paradigm

CBR emerged in the 1980s as an approach to problem solving which offered an alternative to the rule-based approach typical of “expert systems” up until that time. CBR offered a more intuitive approach, based on the way humans appear to solve problems, namely by finding previous similar examples as precedents, and using common-sense reasoning and extrapolation to adapt the precedent to the current problem. This mode of operation is extremely widespread, and can be applied to almost any imaginable human problem. As Leake (1996:3f) states, the CBR approach is based on two tenets about the nature of the world: first, “similar problems have similar solutions”, and second, “future problems are likely to be similar to current problems”. Psychological reality is claimed for CBR as a model of human cognition: “[E]xperts solve problems by applying their experience, whilst only novices attempt to solve problems by applying rules they have recently acquired.” (Watson & Marir, 1994:348)

Riesbeck & Schank (1989) suggest a trade-off between the rule-based and case-based approaches to problem solving: “A rule-based system will be flexible and produce nearly optimal answers, but it will be slow and prone to error. A case-based system will be restricted to variations on known situations and produce approximate answers, but it will be quick and its answers will be grounded in actual experience. In very limited domains, the tradeoffs favor the rule-based reasoner, but the balance changes as domains become more realistically complex.” (p.26)

This methodology applies to a variety of distinct areas of reasoning, and is widely acknowledged as closely modelling human reasoning strategies. In particular, it closely resembles the way human *translators* tend to handle a new text to be translated (Wilss, 1998), which in turn explains the huge popularity among translators of Translation Memory (TM) tools, which are of course a cousin of EBMT (in sharp contrast to the reception that other results of MT research have so far had in that community).

CBR is generally acknowledged to have its roots in Schank & Abelson’s (1977) work on scripts, along with Medin & Schaffer’s (1978) “Exemplar-based Learning”, Stanfill & Waltz’s (1986) “Memory-based Reasoning” and Carbonell’s (1986) “Derivational Analogy”, while the term itself is probably due to Kolodner & Riesbeck (1986).

CBR is often contrasted with rule-based reasoning, in that rules are replaced by cases. By “case” we mean a

“contextualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner” (Kolodner, 1993:13). In fact, cases can be seen as very specific rules, that is, rules which apply to distinct situations. So CBR is a special kind of rule-based reasoning because the rules have to be interpreted “on the fly”, and the same rule may be used differently from one situation to another. Thus far, the same can be said of EBMT.

One of the claimed advantages of CBR is that it overcomes the “knowledge acquisition bottleneck” (Hayes-Roth et al., 1983) of hand-coding a great number of rules, and verifying how they interact with each other. The complexity of real-world domains, according to Riesbeck & Schank (1989:26), makes it “impossible or impractical to specify fully all the rules involved.” In CBR, when the existing “rules” don’t work (i.e. there is no suitable case in the case-base), one simply adds a new one. Such claims have been made for CBR and related techniques (e.g. Watson & Marir, 1994:348) as well as for EBMT itself (e.g. Sato & Nagao, 1990).

Crucial Elements of CBR

Most texts discussing CBR are agreed on the essential elements that make up a CBR system, namely the database of “cases”, of course, and the accompanying design format consisting of the classic “CBR cycle”, as illustrated in Figure 1. A new problem must be appropriately **indexed** to enable suitable past cases to be **retrieved**. These must then be **adapted** to the new problem, and the proposed solution **tested**. If the proposed solution is inappropriate, an **explanation** is usually offered, and **repair** must be attempted; once a correct solution is found, this can be added to the memory, fulfilling a **learning** function.

In the following sections we will take each of these elements in turn to see how they relate to EBMT.

Indexing and Representation

Indexing is the term used in the CBR literature to refer to analysis of the new problem in terms of features relevant to finding and comparing cases, the features being referred to as **indexes**. The indexing scheme affects all the other parts of the system, since it reflects and determines the way the cases are represented, that is, the aspects of the cases which are relevant to the problem domain. Kolodner (1993:145) talks in terms of “the lesson(s) [a case] teaches and the context in which it can teach its lesson(s)”: the lessons are the case’s **content**, the context its indexes.

Since CBR is a problem-solving methodology, the content of a case is often thought of in terms of a **problem description** coupled with its **solution** and, optionally an **outcome**. Some cases may also include an explicit **explanation** of how the solution relates to the description. Typical examples of CBR are a system which tries to find a suitable menu given some ingredients and diners’ preferences and constraints (Hammond, 1986), a medical diagnosis system (Koton, 1988), an agony aunt (Domeshek, 1991). The vocabulary of problem solving permeates the CBR literature: “Cases can be represented in a variety of forms using the full range of AI representational formalisms, including frames, objects, predicates, semantic nets and rules—the frame/object

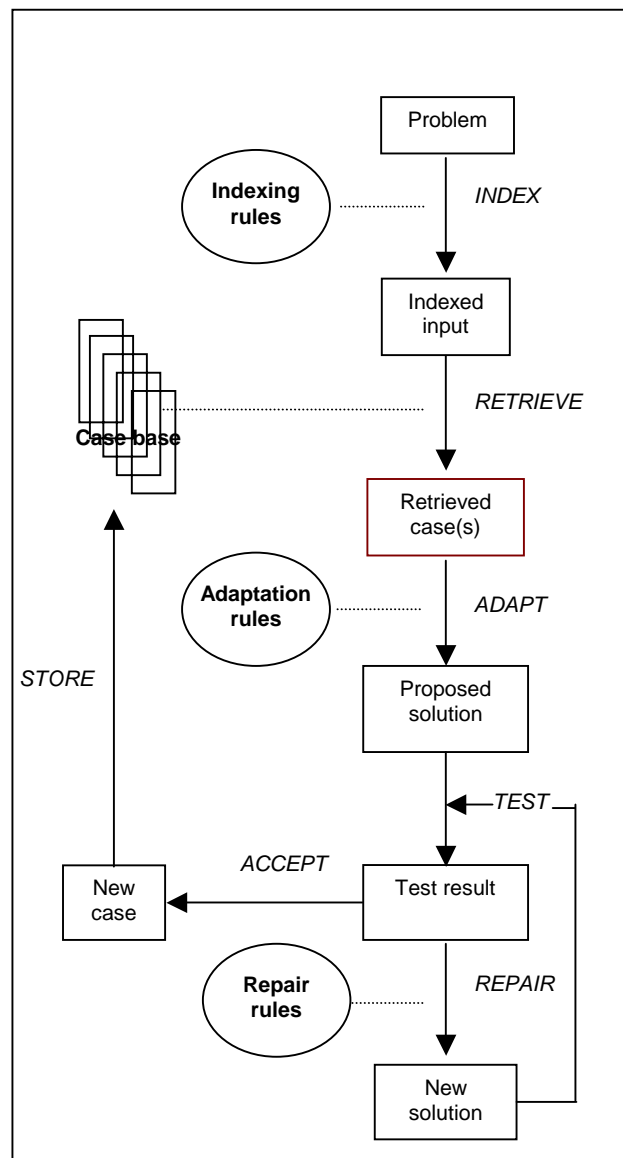


Figure 1. The CBR cycle (based on Riesbeck & Schank, 1989:32)

presentation currently being used by the majority of CBR software.” (Watson & Marir, 1994:331)

Figure 2 shows an example.

Here already we see a difference between CBR and EBMT: many CBR systems address “problems” which have “solutions” which involve a sequence of **goals** to achieve, and “outcomes” or changes to the state of the “world” after the case has been invoked. In contrast, in EBMT the examples are of input–output mappings, and the means of getting from one to the other is rarely made explicit, except inasmuch as elements in the input pattern may be linked to corresponding elements in the output pattern. We will see consequences of this difference at almost every stage.

The content of the cases is distinguished in the literature from the indexing scheme used to retrieve cases. “Indexing” refers specifically to those features of the content that will be used for retrieval. As such they should correspond to the features most likely to describe and distinguish different cases. Kolodner (1993:198ff) suggest that indexes should be predictive, abstract (but not too abstract), and above all useful for the later reasoning task.

```

Problem:
  (include tofu)
  (taste hot)
  (style stir-fry)
Solution:
  (ingredients
  ingr1 (tofu lb .5)
  ingr2 (soy-sauce tablespoon 2)
  ingr3 (rice-wine spoon 1)
  ingr4 (cornstarch tablespoon .5)
  ingr5 (sugar spoon 1)
  ingr6 (broccoli lb 1)
  ingr7 (r-pepper piece 6))
  (actions
  act1 (chop object (ingr1) size
        (chunk))
  act2 (marinate object (result act1)
  in (& (ingr2)(ingr3)(ingr4) (ingr5))
      time (20))
  act3 (chop object (ingr6) size
        (chunk))
  act4 (stir-fry object (& (result
  act2)(ingr7)) time (1))
  act5 (add object (result act3) to
        (result act4))
  act6 (stir-fry object (result act5)
        time (2)))
  (style stir-fry)

```

Figure 2. Example of a case, cited in Kolodner (1993:172)

As in many other aspects of CBR, we should be guided by the intuitive nature of this approach, and consider as indexes the kinds of features that *humans* might naturally select.

So how can we relate this terminology to EBMT? In the EBMT literature, the nature of the case base is widely discussed (cf. Somers, 1999). The cases (examples) are represented in a variety of formats, such as lexically aligned tagged sentences (e.g. Kitano, 1993), tree structures (e.g. Sato & Nagao, 1990; Al-Adhaileh & Kong, 1999), multi-level lattices (e.g. Planas & Furuse, 1999), and so on. Theoretically the cases could “speak for themselves” and be stored as unanalysed pairs of strings, though no EBMT system is reported to take this extreme step. This *is* the case with Translation Memories, a special case of EBMT which, in CBR terms, has the retrieval and storage functions, but leaves adaptation to the human user.

One of the ironies of EBMT is that the mechanisms used to produce the annotations (in CBR terms, indexes) for the cases, and also to analyse a new case into the appropriate format, are usually the same as, or very similar to, the rules found in the rule-based systems they are supposed to replace.

Many of the earliest EBMT systems were hybrids, using the example-based method for only a part of the process (e.g. Sato & Nagao, 1990; Watanabe, 1992; and several others) or for certain problems (e.g. Sumita et al., 1990). Similarly, in many CBR systems the adaptation stage (see below) is rule-based (Leake, 1996:11), while some systems have a rule-based component as a back-up if not relevant cases are available (e.g. Goel et al., 1994; Koton, 1988).

A feature of recent EBMT research has been the tendency to take similar examples and store them as a

single **generalized** example, sometimes so much so that they resemble traditional transfer rules (e.g. Kitano & Higuchi, 1991; Furuse & Iida, 1992; Brown, 1999). Some researchers report procedures for automatically discovering generalized patterns (Cicekli & Güvenir, 1996; Güvenir & Tunç, 1996; Güvenir & Cicekli, 1998; McTait & Trujillo, 1999). The notion of “generalization” is found in the CBR literature, but in a limited way. Riesbeck & Schank (1989:36ff) describe the dynamic formation of “new abstractions ... when a number of cases are discovered to share some common set of features”, and Branting (1989) describes a system which integrates generalizations into a CBR system. Hammond (1989) similarly suggests that abstract cases can be created where common sets of features are shared. Bergmann & Wilke (1996) explore the idea further. On the other hand, Kolodner (1993:7) suggests that generalization, although possible, is not a significant feature of CBR systems. One senses that general rules are in some way the antithesis of the CBR philosophy.

Representation and Retrieval

The way the cases are represented (“indexed”) is of course intimately related to the method of **retrieving** cases which are similar to the given input. This mechanism involves a similarity metric which is used to rank the cases retrieved, together with a search algorithm.

Storage and retrieval is a much-visited problem in CBR. There is a tension between the semantic richness of cases and the efficiency of retrieval, and there is much literature discussing this problem. A striking feature, from the point of view of EBMT, is the small size of the case-base in many of the earliest reported systems: size does not seem to be an issue for CBR, though when it is mentioned at all we see figures in the low hundreds (though cf. Daengdej et al., 1996). In EBMT an example set of under 1,000 would be considered small, and the bigger systems might have as many as three-quarters of a million examples (cf. Somers, 1999:120).

A much used **similarity metric** in CBR is expressed as in (1), where w_i is the importance (“weight”) of feature f_i and s is a similarity function for the individual features, I and R indicating input and retrieved cases, respectively.

$$(1) \quad S(I, R) = \frac{\sum_{i=1}^n w_i \times s(f_i^I, f_i^R)}{\sum_{i=1}^n w_i}$$

There are obviously three elements to this: identification of the primitive similarity function(s) s , determination of the weights w associated with each feature, and an algorithm for finding the case for which the equation in (1) gives the best value.

The **primitive similarity functions** depend on the choice of features and in particular their complexity. Where features map in an obvious way onto scalar ranges, s can involve simple arithmetic. If the features are more complex then correspondingly more complex functions have to be invoked.

More problematic is the assignment of **weights**. One method is simply to ask human experts to do this (or to weight all features equally). More adventurous systems include a component which learns which features are the

most predictive of case differences, or which features are more or less likely to be adapted, and adjusts the relative weights accordingly.

Many CBR systems reportedly use a quite simple search **algorithm** which exhaustively applies the similarity function to all the cases in the case-base. Retrieval time increases linearly with the size of the case-base. One obvious way to overcome this is to organize the search-space so that the higher-weighted features are compared first.

For EBMT, these do not appear to be important issues. Even in early systems where examples are stored as tree structures, little detail is given concerning how tree structures are compared. A similarity metric which makes use of linear distance in a hierarchical thesaurus is widely used for quantifying word similarity (e.g. Nagao, 1984). For the most part, in EBMT the examples have very simple structures, typically sequences of words (this is the case with TMs), or word-tag pairs. The string-edit distance algorithm (Levenshtein, 1966) is widely used, sometimes effectively weighting certain words or categories favourably (e.g. Craniias et al., 1997; Furuse & Iida, 1994; Veale & Way, 1997).

While many CBR and EBMT systems try to retrieve the single best match, or at least to supply a ranking to a set of matches, some systems permit **multiple retrievals**, the idea being that the correct solution will result from taking the best bits of each of them. These might also be described as **partial retrievals**, where the cases are decomposed, making a collection of “substrings” (Nirenburg et al., 1993; Brown, 1997), “fragments” (Somers et al., 1994) or “chunks” (Cunningham et al., 1994; Collins, 1998) of matched material. Figure 3 illustrates this idea.

The idea of using fragments of cases is found in a

```

danger/NN0 of/PRP NN0 < > above/PRP
danger/NN0 of/PRP
of/PRP NN0 < > above/PRP
above/PRP CRD m/NP0
there/PNP is/VVV a/AT0
avalanche/NN0 < > above/PRP
there/PNP is/VVV
is/VVV a/AT0
danger/NN0 of/PRP avalanche/NN0
avalanche/NN0 above/PRP CRD m/NP0
avalanche/NN0 above/PRP
of/PRP avalanche/NN0
there/PNP is/VVV < > a/AT0
is/VVV < > a/AT0
there/PNP is/VVV a/AT0 < > danger/NN0 <
> of/PRP
there/PNP is/VVV < > danger/NN0 < >
of/PRP
there/PNP is/VVV a/AT0 < > danger/NN0
a/AT0 < > danger/NN0
there/PNP is/VVV < > danger/NN0

```

Figure 3. Fragments extracted for the input *there is a danger of avalanche above 2000m.*

The individual words are tagged; the matcher can also match tags only, and can skip unmatched words, shown as < >. The fragments are scored for relevance and frequency, which determines the order of presentation. From Somers et al. (1994).

number of CBR systems, including Redmond (1990), who describes how especially more complex problems can be addressed by looking at subgoals individually, and correspondingly storing cases in “snippets”. Marir & Watson (1995) describe a system to estimate building and refurbishment costs, where the complex problems and solutions are all broken down into “subcases”: the context information becomes all important in this case, since superficially similar solutions can be quite inappropriate if the underlying situation is different.

Adaptation

A solution retrieved from a stored case is almost never exactly the same as a new case. CBR systems therefore need one or more strategies for **adapting** the old system to the new situation. This procedure involves two tasks: first deciding *what* aspect of the stored solution needs to be adapted and then *how* to carry out the adaptation. For many, this adaptive aspect is the heart and soul of CBR. Riesbeck & Schank (1989) refer to it as “the ultimate task of a case-based reasoner” (p. 41). It is important because it not only permits the reuse of existing solutions, but it also contributes to the creation of new solutions and hence to the *learning* capability of CBR.

Despite its importance, adaptation is sometimes omitted from CBR systems, or replaced with human intervention. Watson & Marir comment that “it should not be viewed as a weakness of CBR that it encourages human collaboration in decision support” (1994:330). In CLAVIER (Mark, 1989), an early commercial CBR system, it is reported by Mark et al. (1996) that, as the case-base grew through usage, adaptation and maintenance became more difficult, and eventually the designers decided to replace the adaptation component with an interactive module.

The EBMT equivalent of a system which consists essentially of a retrieval mechanism whose output is then passed to a human is a TM system; and this author (Somers, 1999:114) has explicitly tried to distinguish EBMT and TMs on precisely these grounds: what makes EBMT an interesting process is the extent to which the “hard” part is automated! Similarly, CBR can hardly be considered “reasoning” if its performance amounts to copying and pasting.

The CBR literature is in agreement that adaptation is the most taxing aspect of the paradigm. Hanney & Keane (1997) for example refer to the “adaptation knowledge bottleneck” suggesting that it is difficult to derive any knowledge about how adaptation should be conducted from the cases alone. What is needed is some prior domain knowledge which serves to contextualize the cases; and this domain knowledge is necessarily expressed as general *rules*. In this way, hybrid case- and rule-based systems are developed.

We can see this approach in some EBMT systems, where the cases are “generalized”, as described above, sometimes to such an extent that they really end up as rules.

Overviews (e.g. Riesbeck & Schank, 1989:43; Kolodner, 1993:395ff; Watson & Marir, 1994:334) list up to a dozen types of adaptation, broadly divided between **structural** and **derivational** adaptation techniques. In the former, rules are applied to (a copy of) the case selected as the best match. In the latter, the algorithms, methods or

rules which generated the original solution are reused to generate a new solution. Because of differing terminology, it is not always clear whether differently named methods are really distinct.

The adaptation step in EBMT is usually termed **recombination**, though this term is more specifically applicable in systems where the matching process retrieves multiple, sometimes partial, solutions, a strategy not widely used in CBR.

The simplest of the CBR adaptation methods is null adaptation; then there are substitution methods (reinstantiation, parameter adjustment, abstraction and respecialization, case-based substitution, specialized search), transformation methods (commonsense transformation and model-guided repair) and finally derivational replay.

Null adaptation

The first method in reality involves no adaptation at all. Clearly it is used when the new problem exactly matches an existing case, or it may be used when the new problem is sufficiently close to the matched case that no adaptation is necessary (bearing in mind the existence of a revision stage). In EBMT, null adaptation occurs when an exact match is found, which may be more or less common depending on the application, but for null adaptation to apply when the match is not exact would involve the system “knowing” that the differences between the input and the match were insignificant. One can imagine ways of doing this.

Reinstantiation

In reinstantiation, the old and new problems are structurally similar, but differ in the values of elements. Reinstantiation involves replacing the old values with new. This is a method often found in EBMT: for instance (2) could be used as a model for the translation of (3) by replacing *she* with *he*, *big* with *blue* and *feet* with *eyes* to give (4).

- (2) *Kanojo wa ashi ga ōkii.*
SHE topic FOOT subj BIG
She has big feet.
- (3) *Kare wa me ga aoi.*
HE topic EYE subj BLUE
- (4) He has blue eyes.

In reinstantiation we have to know the correspondences between the elements that we are exchanging, but we also have to be sure that the simple substitution is permitted. In CBR terms, if there are implicit relationships between the slots, reinstantiating one might have repercussions. This can be easily illustrated in the case of EBMT: if we want to use (5) as a model for the translation of (6), we cannot simply replace *man-homme* with *woman-femme* (7), but also change some of other words in the sentence (7).

- (5) That old man has died.
Ce vieil homme est mort.
- (6) That old woman has died.
- (7) a. **Ce vieil femme est mort.*
b. *Cette vieille femme est morte.*

This problem is referred to in the EBMT literature as **boundary friction** (Nirenburg et al., 1993:48; Collins,

1998:22; Somers, 1999:133). One solution to this problem in CBR terms might be to treat it as a case for ...

Parameter adjustment

This is a structural adaptation technique in which specific parameters of the retrieved and new cases differ. A key element seems to be the use of “specialized adjustment heuristics” to cope with the problem (Kolodner, 1993:404). A possible interpretation of this in EBMT terms is if in (5) the representation of the French translation included an indication of the agreement requirements, so that the substitution of *man* with *woman* in (6) would trigger specific agreement rules to adapt the other words. MT experts might call such a specialized adjustment heuristic a “transfer rule”.

Abstraction and respecialization

This technique, also termed “local search”, is a type of substitution that allows a novel solution to be generated from an example which differs only in a small part. The idea is to take the piece of the solution that does not fit, look for abstractions of that piece, and then try other specializations of the abstraction in the current situation. This technique obviously depends on there being a hierarchically structured knowledge base behind the case base, and is very well illustrated for EBMT by Sumita & Iida’s (1991) system which translates Japanese adnominal particle constructions (*A no B*) with the help of a thesaurus.

Case-based substitution

This adaptation technique comes into play when parts of a solution have to be found in additional cases.

Papaioannou (2000) adopted this as a solution to the boundary friction problem in a simulated English–Greek EBMT system. Because Greek is a highly inflected language, there is danger of recombining inappropriately inflected fragments. The examples in Papaioannou’s system are tagged and lemmatised to show appropriate morphological information, as in Figure 4.

```
I saw the new prime-minister.
Είδα τον νέο πρωθυπουργό.
<s>
  <gw cat="V" attrs="Act Pst Ind Sng 1st"
    lemma="blepw/see">Eida</gw>
  <gw cat="Art" attrs="Msc Sng Acc"
    lemma="o/the">ton</gw>
  <gw cat="Adj" attrs="Msc Sng Acc"
    lemma="neos/new">neo</gw>
  <gw cat="N" attrs="Msc Sng Acc"
    lemma="prw8upourg0s/prime-
    minister">prw8upurgo</gw>
  <punc>.</punc>
</s>
```

Figure 4. An example from Papaioannou (2000).

The recombination stage “knows” that certain attributes have to match up (agreement of Art, Adj and N, for instance), so when the system retrieves examples for a new input, it notes the particular details of any discrepancies and specifically searches the rest of the example-base for the missing item(s). For instance, the sentence in (8) matches the example in Figure 4 perfectly,

except for the last lemma and surface word. In order to adapt the example, the system searches the example-base for another case that contains exactly the configuration in (9), and, if it is found can generate the appropriate form *πρόεδρο*.

(8) I saw the new president.

(9) <gw cat="N" attrs="Msc Sng Acc" lemma=" ???/president">??? </gw>

If we now give the input (10), there are several mismatches. Two of the words have the wrong attributes, and the third word also has the wrong lemma. So the system has to search for the three items in (11).

(10) I saw the new delegates.

(11) a. <gw cat="Art" attrs="Msc Plr Acc" lemma=" o/the">??? </gw>

b. <gw cat="Adj" attrs="Msc Plr Acc" lemma=" neos/new">??? </gw>

c. <gw cat="N" attrs="Msc Plr Acc" lemma=" ???/ delegate">??? </gw>

Supposing there is no “evidence” in the case-base for one of these new combinations. If the missing case is (11), where we do not know the lemma, there is not much we can do. In the case of (11), we may be able to generate the appropriate form of the adjective by looking at other masculine plural adjectives, and comparing the lemma and the surface form, though this would be a further complexity for the adaptation phase. This might be termed “specialized search” (Kolodner, 1993:411).

Common-sense transformation

Kolodner (1993) describes two types of adaptation involving “transformations”. Transformation in general involves making deletions or insertions either to the solution as a whole or to some part of it. The first of these is common-sense transformation, which makes use of “a small set of heuristics that use knowledge about the relative importance of different components of an item to determine whether deletions or substitutions should be done” (p. 420f). To be able to do this, the system must of course identify the component needing to be changed, but the representations need to indicate which components are susceptible to this kind of manipulation. In particular, the internal relationships between the components must be maintained after the transformation has taken place.

How might this be implemented in an EBMT system? The idea of deleting or inserting components is widespread in EBMT systems, and is very intuitive. If we have the Malay–English examples in (12), it is not difficult to construct the correct translations of sentences like those in (13).

(12) a. *Dia nak pergi ke kedai beli roti.*

She is going to go to the shops to buy bread.

b. *Dia pergi ke pasar nak beli baju.*

She went to the market to buy a shirt.

c. *Mereka pergi ke kampung nak beli kereta.*

They went to the village to buy a car.

(13) a. She went to the village to buy bread.

b. They are going to the market.

In fact as humans we bring to bear a certain amount of generic (common-sense) knowledge about how languages work to do this. The work – mentioned above – to extract

patterns fully automatically (Cicekli & Güvenir, 1996; Güvenir & Tunç, 1996; Güvenir & Cicekli, 1998; McTait & Trujillo, 1999) needs minimal pairs and would not be able to extract as much as humans can from the examples in (12).

This kind of activity does have its limitations though. Kolodner (1993) notes that internal relationships between elements must be maintained, and here we meet again the “boundary friction” problem already illustrated in (5)–(7). A further problem is that language is not always logical: you might guess from the examples in (14)

(14) *stryd fawr* – big street

stryd fach – small street

tŷ mawr – big house

??? – small house

that the Welsh for ‘small house’ is *tŷ mach* ... unfortunately though you would be wrong (it should be *bach*!).

Model-guided repair

For these reasons, model-guided repair might be a better way of implementing adaptation. As its name suggests, the transformations are guided by some knowledge of the domain, rather than just common sense. In EBMT terms, this would mean verifying that the proposed adaptation is legitimate by submitting it to some kind of analysis. As we will see below, most CBR systems involve an evaluation step where this kind of verification is done, but it could be regarded as part of the adaptation procedure. An obvious way to verify a proposed translation is to try to parse it, though this step would require a grammar of the target language which, perhaps, would undermine the reason for adopting the example-based approach in the first place.

Somers et al. (1994) proposed “disalignment”: the proposed output was checked by rerunning the matching algorithm that had been used on the input, this time comparing the proposed output against the target side of the examples. The ease with which the proposed output could be matched was felt to be an indication of its well-formedness.

Derivational replay

Kolodner (1993) describes “derivational replay” as follows:

Each of the adaptation methods described up to now fixes an old solution to fit a new solution. Sometimes, however, it is more appropriate to recompute a new solution or partial solution using the same means by which the old solution was computed. (p. 435)

A problem with seeing how this technique could apply to EBMT is that there is an underlying assumption that solutions are “computed”, whereas in EBMT the solutions are usually just “given”. Where solutions (i.e. translations) are computed, this is usually in the traditional rule-based manner. So we could regard the hybrid systems that have been described in this light. In the early days of EBMT it was sometimes suggested that this method could be used just for special problem cases (cf. Sumita et al., 1990).

Adaptation-guided retrieval

An implicit assumption in many CBR systems is that the most similar case in the casebase is also the most adaptable. This is not always the case, as Smyth & Keane

(1993, 1995), Leake (1995), Rousen & Aarts (1996) and others have noted. The notion of “adaptation-guided retrieval” has been developed whereby case retrieval is based not only their similarity with the given input, but also the extent to which they represent a good model for the desired output, i.e. to which they can be adapted, that determines whether they are chosen. Collins (1998:31) gives the example of a robot using a restaurant script to get food at Macdonald’s, when buying a stamp at the post-office might actually be a more appropriate, i.e. adaptable, model. Collins et al.’s EBMT system, ReVerb, stores the examples together with a functional annotation, cross-linked to indicate both lexical and functional equivalence. This means that example-retrieval can be scored on two counts: (a) the closeness of the match between the input text and the example, and (b) the adaptability of the example, on the basis of the relationship between the representations of the example and its translation. Obviously, good scores on both (a) and (b) give the best combination of retrievability and adaptability, but we might also find examples which are easy to retrieve but difficult to adapt (and are therefore bad examples), or the converse, in which case the good adaptability should compensate for the high retrieval cost.

Evaluation and Repair

Once a solution has been proposed by the case-based reasoner, the system should then, according to many commentators, **evaluate** the proposal and, if necessary, **revise** (or repair) it. An important part of this is explanation, whereby the reasoner gives the reasons for the failure of the solution. An obvious and important element here is that the system should apparently be aware that it has failed to provide a correct solution – the “test” phase shown in Figure 1 above.

The language in which these ideas are expressed, for example in Schank & Riesbeck (1989) once again reflects the problem-solving scenario that so strongly influences researchers’ ideas in CBR. They suggest that two sorts of failures are possible: “Goals specified in the input are simply not achieved ... [or] implicit goals, not specified in the input, are violated” (p. 52). Nevertheless, it quickly emerges from study of the literature that any “evaluation” of the system’s output will be done by human intervention. Any alternative introduces a familiar vicious circle: if the system knew that it had got the wrong answer, it wouldn’t have got the wrong answer.

On the other hand, one of the advantages of empirical approaches (as opposed to rule-based approaches) is the role that stochastic processing can play. Empirical methods can deliver a “certainty rating” along with any result, inasmuch as the result is based on probabilities. Thus, it is claimed, an EBMT system for example not only tells you what it thinks the translation is, but how sure it is of that answer, what its second and third choice solutions might be, and how far behind they are in the ranking.

One area that is important in the CBR community, but which seems to be lacking in EBMT, is **explanation**. Originally thought of as a step in the CBR cycle (see Figure 1 again) to facilitate repair of faulty solutions prior to learning, it has developed into a major field of its own (cf. Schank et al., 1994) related to AI tasks such as story understanding.

All in all, however, it seems that for EBMT, with its shallow representation of cases, and typically uncomplicated “reasoning”, there is not much on offer here.

Learning for Reuse

The final step in CBR is to store accepted solutions in the case-base for later use. In the same way, an approved translation can be added to the example-base. This is normal practise in the case of TMs, where the whole process is supervised by a human user. Descriptions of EBMT systems do not as a rule mention this possibility, nor is it ruled out.

The main issue here is the need for human quality control, though, conversely, some TM systems allow examples to be stored with an annotation identifying whether their source is human translation or MT (including EBMT).

Kolodner (1993:566) briefly discusses some of the issues in case-based learning. She notes that new cases can be added by the system itself and, of course manually. She also suggests that storing failures, i.e. cases where the system could not come up with the correct solution, can be beneficial: a system can learn from its mistakes as well as from its successes, especially if the failed case is annotated to show where it went wrong. There may be something in this for EBMT.

A second issue is the effect on the case-base of storing multiple similar or even identical examples. As Kolodner (1993) states, “Some cases added to a case library might add to the library without adding to the capabilities of a system. Such cases might or might not affect the efficiency of retrieval, depending on the retrieval algorithm.” (p. 567)

A similar point can be made with regard to EBMT (see Somers, 1999:121). If we construct the example-base from a corpus, there will be repetition and overlap. Some examples will mutually reinforce each other, being identical, or exemplifying the same translation phenomenon. But other examples will be in conflict: the same or similar phrase in one language may have two different translations for no other reason than inconsistency (e.g. Carl & Hansen, 1999:619).

Where examples reinforce each other this may or may not be useful. Some systems (e.g. Somers et al., 1994; Öz & Cicekli, 1998; Murata et al., 1999) involve a similarity metric which is sensitive to frequency. But if no such weighting is used, multiple similar or identical examples are just extra baggage, introducing a kind of spurious ambiguity because there seem to be multiple alternative solutions to the same problem.

Conclusions

The aim of this essay was to see if the similarity between EBMT and CBR was anything more than skin-deep. My hope was to find some inspiration in the CBR literature to push EBMT forward.

I think we have shown that in many respects EBMT can be described using CBR terminology, but for the most part we do not learn anything from this. The main problem seems to be the discrepancy in the complexity of representation of examples in CBR compared to EBMT. If anything, the CBR community might find EBMT an

interesting special case, showing what one can do with very superficial analysis and representation of the cases.

I think the most fruitful area for further consideration is the question of adaptation, especially the way it might interact with retrieval, as the Dublin group (Collins and colleagues) have already shown us.

References

- Al-Adhaileh, M.H. and E.K. Tang. 1999. Example-based machine translation based on the synchronous SSTC annotation schema. *Machine Translation Summit VII*, Singapore, pp. 244–249.
- Bergmann, R. and W. Wilke. 1996. On the role of abstraction in case-based reasoning. In *EWCBR-96*, pp. 28–43.
- Branting, L.K. 1989. Integrating generalizations with exemplar-based reasoning. *Case-Based Reasoning: Proceedings of a Workshop on Case-Based Reasoning*, Pensacola Beach, Florida, pp. 224–229.
- Brown, R.D. 1997. Automated dictionary extraction for “knowledge-free” example-based translation. *Proceedings of the 7th International Conference on Theoretical and Methodological Issues in Machine Translation*, Santa Fe, New Mexico, pp. 111–118.
- Brown, R.D. 1999. Adding linguistic knowledge to a lexical example-based translation system. *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 99)*, Chester, England, pp. 22–32.
- Carbonell, J.G. 1986. Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In R. Michalski, J. Carbonell and T. Mitchell (eds) *Machine Learning: An Artificial Intelligence Approach*, Vol. 2, pp. 371–392.
- Carl, M. and S. Hansen. 1999. Linking translation memories with example-based machine translation. *Machine Translation Summit VII*, Singapore, pp. 617–624.
- Cicekli, I., and H.A. Güvenir. 1996. Learning translation rules from a bilingual corpus. *NeMLaP-2: Proceedings of the Second International Conference on New Methods in Language Processing*, Ankara, Turkey, pp. 90–97.
- Collins, B. 1998. *Example-based Machine Translation: an adaptation-guided retrieval approach*. PhD thesis, Trinity College, Dublin.
- Collins, B. and P. Cunningham. 1995. A methodology for example-based machine translation. *CSNLP 1995: 4th Conference on the Cognitive Science of Natural Language Processing*, Dublin.
- Collins, B. and P. Cunningham. 1996. Adaptation-guided retrieval in EBMT: A case-based approach to machine translation. In *EWCBR-96*, pp. 91–104.
- Collins, B. and P. Cunningham. 1997. Adaptation guided retrieval: Approaching EBMT with caution. *Proceedings of the 7th International Conference on Theoretical and Methodological Issues in Machine Translation*, Santa Fe, New Mexico, pp. 119–126.
- Collins, B., P. Cunningham and T. Veale. 1996. An example-based approach to machine translation. *Expanding MT Horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas*, Montreal, Quebec, pp. 1–13.
- Cranias, L., H. Papageorgiou and S. Piperidis. 1997. Example retrieval from a translation memory. *Natural Language Engineering* 3, 255–277.
- Cunningham, P., B. Smyth and T. Veale. 1994. On the limitations of memory based reasoning. In *EWCBR-94*, pp. 75–86.
- Daengdej, J., D. Lukose, E. Tsui, P. Beinat and L. Prophet. 1996. Dynamically creating indices for two million cases: a real world problem. In *EWCBR-96*, pp. 195–119.
- Domeshek, E.A. 1991. What Abby cares about. *Case-Based Reasoning: Proceedings of a Workshop...*, Washington, D.C., pp. 13–24.
- EWCBR-93* = S. Wess, K.-D. Althoff and M.M. Richter (eds), *Topics in Case-Based Reasoning: First European Workshop*, *EWCBR-93*, Kaiserslautern, Germany... (Lecture Notes in Computer Science 837), Berlin: Springer
- EWCBR-94* = J.-P. Haton, M. Keane and M. Manago (eds) *Advances in Case-Based Reasoning: Second European Workshop*, *EWCBR-94*, Chantilly, France... (Lecture Notes in Computer Science 984), Berlin: Springer.
- EWCBR-96* = I. Smith and B. Faltings (eds), *Advances in Case-Based Reasoning: Third European Workshop*, *EWCBR-96*, Lausanne, Switzerland... (Lecture Notes in Computer Science 1168), Berlin: Springer.
- Furuse, O. and H. Iida. 1992. An example-based method for transfer-driven machine translation. *Quatrième colloque international sur les aspects théoriques et méthodologiques de la traduction automatique; Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*. TMI-92, Montréal, Québec, pp. 139–150.
- Furuse, O. and H. Iida. 1994. Constituent boundary parsing for example-based machine translation. *COLING 94, The 15th International Conference on Computational Linguistics*, Kyoto, Japan, pp. 105–111.
- Goel, A., K. Ali, M. Donnellan, A. Garza and T. Callantine. 1994. Multistrategy adaptive navigational path planning. *IEEE Expert*, 9.6, 57–65.
- Güvenir, H.A. and I. Cicekli. 1998. Learning translation templates from examples. *Information Systems* 23, 353–363.
- Hammond, K.J. 1986. CHEF: a model of case-based planning. *Proceedings AAAI-86 Fifth National Conference on Artificial Intelligence*, Philadelphia, Pa., pp. 267–271.
- Hammond, K.J. 1989. On functionally motivated vocabularies: an apologia. *Proceedings of the International Joint Conference on Artificial Intelligence*, Milan, Italy.
- Hanney, K. and M.T. Keane. 1997. The adaptation knowledge bottleneck: How to ease it by learning from cases. In *ICCBR-97*, pp. 359–370.
- Hayes-Roth, F., D. Waterman and D. Lenat (eds) 1983. *Building Expert Systems*. Reading, Mass.: Addison Wesley.
- ICCBR-95* = M. Veloso and A. Aamodt (eds), *Case-Based Reasoning Research and Development: First International Conference on Case-Based Reasoning*, *ICCBR-95*, Sesimbra, Portugal... (Lecture Notes in Computer Science 1010), Berlin: Springer
- ICCBR-97* = D.B. Leake and E. Plaza (eds), *Case-Based Reasoning Research and Development: Second*

- International Conference on Case-Based Reasoning, ICCBR-97, Providence, RI ...* (Lecture Notes in Computer Science 1266), Berlin: Springer.
- Kitano, H. 1993. A comprehensive and practical model of memory-based machine translation. *Proceedings of the International Joint Conference on Artificial Intelligence*, Chambéry, France, pp. 1276–1282.
- Kitano, H. and T. Higuchi. 1991. High performance memory-based translation on IXM2 massively parallel associative memory processor. *AAAI-91 Proceedings Ninth National Conference on Artificial Intelligence*, Menlo Park: AAAI Press/The MIT Press, pp. 149–154.
- Kolodner, J. 1993. *Case-Based Reasoning*. San Mateo, CA: Morgan Kaufmann.
- Kolodner, J.L. and D.B. Leake. 1996. A tutorial introduction to case-based reasoning. In Leake (1996), pp. 31–65.
- Kolodner, J.L. and C.K. Riesbeck. 1986. *Experience, Memory and Reasoning*. Hillsdale, New Jersey: Laurence Erlbaum.
- Koton, P. 1988. Reasoning about evidence in causal explanation. *AAAI 88 The Seventh National Conference on Artificial Intelligence*, St Paul, Minnesota.
- Leake, D. 1995. Adaptive similarity assessment for case-based explanation. *International Journal of Expert Systems* **8**, 165–194.
- Leake, D.B. (ed.) 1996. *Case-based Reasoning – Experiences, Lessons & Future Directions*. Menlo Park, California: AAAI Press / MIT Press.
- Levenshtein, V.I. (1966) Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, **10**, 707–710.
- Marir, F. and I. Watson. 1995. Representing and indexing building refurbishment cases for multiple retrieval of adaptable pieces of cases. In *ICCB-95*, pp. 55–66.
- Mark, W.S. 1989. Case-based reasoning for autoclave management. *Case-Based Reasoning: Proceedings of a Workshop on Case-Based Reasoning*, Pensacola Beach, Florida, pp. 176–180.
- Mark, W.S., E. Simoudis and D. Hinkle. 1996. Case-based reasoning: expectations and results. In Leake (1996), pp. 267–294.
- McTait, K. and A. Trujillo. 1999. A language-neutral sparse-data algorithm for extracting translation patterns. *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 99)*, Chester, England, pp. 98–108.
- Medin, D.L. and M.M. Schaffer. 1978. Context theory of classification learning. *Psychological Review* **85**, 207–238.
- Murata, M., Q. Ma, K. Uchimoto and H. Isahara. 1999. An example-based approach to Japanese-to-English translation of tense, aspect, and modality. *Proceedings of the 8th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 99)*, Chester, England, pp. 66–76.
- Nirenburg, S., C. Domashnev and D.J. Grannes. 1993. Two approaches to matching in example-based machine translation. *Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation TMI '93: MT in the Next Generation*, Kyoto, Japan, pp. 47–57.
- Öz, Z. and I. Cicekli. 1998. Ordering translation templates by assigning confidence factors. In D. Farwell, L. Gerber and E. Hovy (eds) *Machine Translation and the Information Soup: Third Conference of the Association for Machine Translation in the Americas, AMTA '98, Langhorne, Pa...* (Lecture Notes in Artificial Intelligence 1529), Berlin: Springer, pp. 51–61.
- Papioannou, V. 2000. *Recombination in Example-based MT: A Case-Based Reasoning approach*. MSc dissertation, Department of Language Engineering, UMIST, Manchester.
- Planas, E. and O. Furuse. 1999. Formalizing translation memories. *Machine Translation Summit VII*, Singapore, pp. 331–339.
- Redmond, M. 1990. Distributed cases for case-based reasoning; facilitating use of multiple cases. *AAAI-90 Proceedings Eighth National Conference on Artificial Intelligence*, Menlo Park: AAAI Press / The MIT Press, pp. 304–309.
- Riesbeck, C.K. and R.C. Schank. 1989. *Inside Case-based Reasoning*. Hillsdale, New Jersey: Lawrence Erlbaum.
- Rousen, J. and R.J. Aarts. 1996. Adaptation cost as a criterion for solution evaluation. In *EWCBR-96*, pp. 354–361.
- Sato, S. and M. Nagao. 1990. Toward memory-based translation. *COLING-90, Papers Presented to the 13th International Conference on Computational Linguistics*, Helsinki, Finland, Vol. 3, pp. 247–252.
- Schank, R.C. and R.P. Abelson. 1977. *Scripts, Plans, Goals and Understanding*. Hillsdale, New Jersey: Lawrence Erlbaum.
- Schank, R.C., A. Kass and C.K. Riesbeck. 1994. *Inside Case-Based Explanation*. Hillsdale, New Jersey: Lawrence Erlbaum.
- Skousen, R. 1989. *Analogical Modeling*. Dordrecht: Kluwer.
- Smyth, B. and M.T. Keane. 1993. Retrieving adaptable cases. In *EWCBR-93*, pp. 209–220.
- Smyth, B. and M.T. Keane. 1995. Experiments on adaptation-guided retrieval in case-based design. In *ICCB-95*, pp. 313–324.
- Somers, H. 1999. Review article: Example-based Machine Translation. *Machine Translation* **14**, 113–157.
- Somers, H., I. McLean and D. Jones. 1994. Experiments in multilingual example-based generation. *CSNLP 1994: 3rd Conference on the Cognitive Science of Natural Language Processing*, Dublin, Ireland.
- Stanfill, C. and D. Waltz. 1986. Toward memory-based reasoning. *Communications of the ACM* **29**, 185–192.
- Sumita, E. and H. Iida. 1991. Experiments and prospects of example-based machine translation. *29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, California, pp. 185–192.
- Sumita, E., H. Iida and H. Kohyama. 1990. Translating with examples: a new approach to machine translation. *The Third International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, Austin, Texas, pp. 203–212.
- Veale, T. and A. Way. 1997. *Gaijin: A bootstrapping approach to example-based machine translation. International Conference, Recent Advances in Natural Language Processing*, Tzigov Chark, Bulgaria, pp. 239–244.
- Watanabe, H. 1992. A similarity-driven transfer system. *Proceedings of the fifteenth [sic] International Conference on Computational Linguistics, Actes du quinzième colloque international en linguistique*

informatique, COLING-92, Nantes, France, pp. 770–776.

Watson, I.D. (ed.) 1995. *Progress in Case-based Reasoning: First United Kingdom Workshop, Salford, UK...* (Lecture Notes in Computer Science 1020), Berlin: Springer Verlag.

Watson, I. and F. Marir. 1994. Case-based reasoning: A review. *The Knowledge Engineering Review* **9**, 327–354.

Wilss, W. 1998. Decision making in translation. In M. Baker (ed.) *Routledge Encyclopedia of Translation Studies*, London: Routledge, 57–60.