

# Adaptation Guided Retrieval: Approaching EBMT with Caution

Bróna Collins, Pádraig Cunningham

Department of Computer Science  
Trinity College Dublin  
Ireland  
brona.collins@cs.tcd.ie

**Abstract.** Many translation memory tools rely on the assumption that source language constructions correspond more or less isomorphically to their target language counterparts. Cross-linguistic divergences and non-linguistic factors such as translator stylistics clearly make the assumption false, so there is a need for sub-sentential linkage between any two languages. The strength of these links determines how *adaptable* a particular translation is in the context of a given problem sentence, and we use this as our main selection criterion. Our cautious approach to EBMT is inspired by current adaptation-guided methodologies shown to be successful in other case based reasoning domains.

## 1 Introduction

The idea of reusing the product of previous mental effort is universally attractive, and nowhere is the motivation so great as in the area of translation of repetitive documents. One way of avoiding the wasteful tedium of translating words and phrases over and over again is to maintain a terminology database, which commercial companies have been doing for years. A more principled approach to translation reuse involves devising some methodology for storing and retrieving previous translations, and such research systems come under the umbrella term of Example-Based MT (henceforth EBMT). The flux of EBMT methodologies in the nineties has coincided with the coming to age of a sub-field of AI known as Case-Based Reasoning (henceforth CBR). The basic tenet of CBR is that previous problem-solving knowledge can be stored in such a fashion that it may be reused in order to solve some new problem, assuming a certain degree of similarity between the old and the new problem. The beauty of the notion of reuse is that of avoiding the first-principles approach to any problem and specifically in MT, this means avoiding the “parse, transfer, parse” process. In CBR, the name *Adaptation Guided Retrieval* (henceforth AGR) is given to the notion of retrieving old solutions cautiously, depending on whether they are adaptable or not with respect to the given problem. This policy has been shown to be successful in other domains by [Smyth & Keane, 1995] and [Leake et al, 1995]. In this paper we seek to demonstrate that AGR is a good policy to adopt for the EBMT task in “real-world” domains. In addition, we demonstrate how the adaptability of a previous translation can be quantified.

In Section 2, we outline the problems posed by reusing old translations. This motivates our data-oriented methodology which is loyal to the heart of CBR and EBMT. As an EBMT system is only as good as its data, Section 3 is devoted to justifying how examples are represented in our example base. Section 4 describes how this data is used at run-time and the test results. Section 5 concludes with a discussion of the results of AGR and the use of system lexicons for sub-sentential alignment in general.

## 2 Adaptability of Previous Translations

A previous translation is deemed reusable if its source-language (SL) part is similar enough to the input SL sentence. What is generally meant by *similar enough* is that the differences between the input problem and the retrieved example can be compensated for by adjusting the example SL in such a way that the correspondingly adjusted TL solution is still grammatical and captures the meaning of the SL. The snag is that even though it may be trivial to adapt an example to suit the new problem at hand, this does not guarantee that the corresponding changes can be propagated across to the target language solution. What is required is some contrastive linguistic knowledge to guide the adaptation of the old translation in the context of the new input sentence. Consider the data sample in 1) below, taken from our corpus.<sup>1</sup> Here *sl'* is the new sentence to be translated, *sl* is a previous SL sentence coupled with its translation *tl* in the example-base, and *tl'* is the result of the naive re-use of the old translation to translate the new input:

- 1)  
sl' Use the Zoom Tool to bring all objects into view.  
sl Use the Offset Command to specify the shaping between the shapes.  
tl Mit der Option Abstand legen Sie den Abstand zwischen den Formen fest.  
\*tl' Mit der Option "Zoom" des Hilfsmittel Zoom bringen Sie alle Objekte in den sichtbaren Bereich fest.

The verbs in the sub-ordinate clauses of *sl'* and *sl* appear as infinitives in both, but are translated as the separable verb *festlegen* in *tl'* and the non-separable verb *bringen* in *tl* respectively. In order to correctly reuse the translation function of *sl*→*tl* to translate *sl'*, it is clear that extra knowledge to the effect that *festlegen* is separable, must be available to the system at adaptation time. The actual transfer rule in this case is trivial enough but consider the effect of multiple, interacting transfer rules, see [Streiter & Schmidt-Wigger, 1995] for discussion. Appealing to linguistic and/or domain knowledge causes a knowledge-acquisition bottleneck, see [Sumita et al., 1993], and would seem to contradict the basic motivation behind using an example-based approach in the first place. What a system could do instead, is use the data stored in the example base to ascertain that a certain SL→TL link (e.g. *to specify*) is tenuous because its corresponding target chunk has somehow been split up, and prevent such a reminding being made by penalising the example link. So, when the system tries to translate an input string which differs at this position from the example, it would be forced to consider another example, even one with perhaps less overall similarity to the input, but which is lexically similar in the position where *to specify* appears. By predicting adaptability for sections of a previous translation, and integrating this knowledge into the retrieval process, we have a technique for avoiding cases which require "dangerous" adaptation.

### 2.1 Terminology for Adaptation-Guided EBMT

We adopt the following scheme for naming the components of our adaptation-guided methodology. A case in the case base embodies information about a previous translation, which can be reused, see Figure 1. The example translation consists of a bilingual sentence pair in their original forms SL and TL respectively, and in their abstracted versions, SLA and TLA respectively. A new input problem is denoted SL' and its abstracted form is SLA'. Let us call the function

---

<sup>1</sup> Our bilingual corpus is the English and German Corel DRAW 6 manual, which was aligned manually at the sentence level

transferring the example SLA into the example TLA the original translation function. The modified translation function is then a function of the original translation and the differences

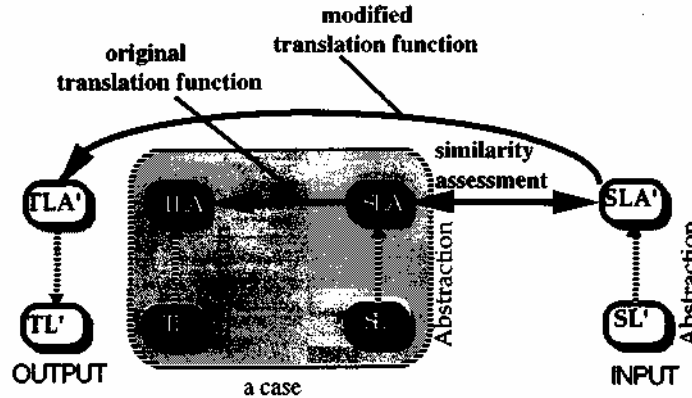


Figure 1. Terminology for Adaptation Guided EBMT

### 3 Creating Adaptation Knowledge

In most corpus-based EBMT systems, the process of matching is seen as being a separate task from that of adaptation, and the differences perceived by the matcher sometimes cannot be incorporated into the adaptation process at all. This is because fragments of decomposed cases are matched and then stitched together to produce a final template from which to generate the TL. The fragments originate from different contexts so it is difficult to have a compositional account of what changes need to be made to the resultant target language string, see [McLean et al., 1994] or [Brown & Frederking, 1995] for a discussion on the boundary friction problem in corpus-based systems.

We suggest that the key to resolving the run-time problem of adapting examples is, firstly, to disallow full case decomposition and, secondly, to couple the retrieval and adaptation processes tightly together, effectively decreasing the workload of the adaptation component. Without complete case fragmentation we run the risk of reduced coverage, but this can be compensated for by representing sentences in abstract template form containing variables in those positions where the SL  $\rightarrow$  TL links were strong. Then, by allowing the adaptor only to adapt variabilised positions of a string, structure-modifying adaptation can be avoided. So, at run time, the retrieval mechanism only passes adaptable cases to the adaptor, - those which it "knows" it can adapt by simple corpus-based substitution of text chunks in positions which have been variabilised. In the following sections we describe how we encode the knowledge required for adaptation guided retrieval in the cases.

### 3.1 Alignment in ReVerb

The REVERB system comprises the case base of English→German examples, where each case is represented in the frame-based knowledge representation language KRELL and the case-linker, case-retriever and case-adaptor. We use an external tool, namely the English constraint grammar ENGCG [Voutilainen, 1995] to parse the SL to a flat syntactic dependency level, and a simple german parser of our own to parse the TL. Then REVERB attempts to link the resultant "chunks" and assign weights to those links. For linking purposes, the system was bootstrapped by inserting some 214 hand-coded cases (sentences) into its memory which incorporated approximately 2,300 linked SLA→TLA chunks. The REVERB linking algorithm works as follows:

```

For every wordi ∈ SL:
  determine csi, the SL chunk that contains it.
  create a list Ti of its most probable translations
  for every tj ∈ Ti
    find set CTj of chunks of TL which contain tj.
  for every ctk ∈ CTj
    increment score(link(csi,ctk))
  
```

This creates multiple links (biased in the direction SL→TL) between chunks where words from  $n$  chunks in one language map to more or less than  $n$  chunks in the other. A diagrammatic view of the linking process is provided in Figure 2 below. In Figure 2a), the words are linked by looking up the system dictionary.

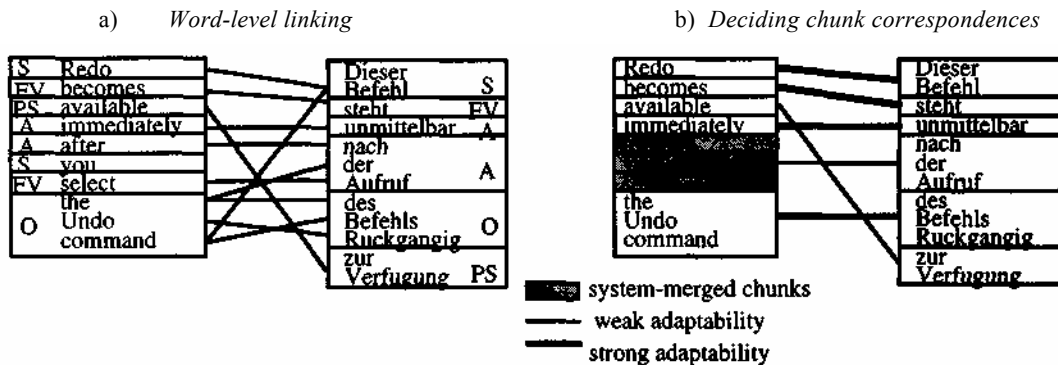


Figure 2. Determining SLA→TLA links automatically in a two-stage process

Notice how the customised system dictionary links *Redo* and *Dieser Befehl* ("this command"). Because the dictionary is made up of cases, it is customised to this particular data-set, in which the word *Redo* must have occurred a sufficiently high number of times with *Befehl* in order to render it a likely translation. See Section 4.2 for further discussion. The symbols at the chunk perimeters indicate the flattened dependency-structures as output by the ENGCG grammar, for example: S:subject, FV: Finite main Verb. In order to establish a one-to-one mapping between the SLA and TLA chunks, the following spreading-activation scheme is employed:

For every link  $(cs_i, ct_j)$   
determine set  $CT_i$  of target chunks that are linked to  $cs_i$   
determine set  $CS_j$  of source chunks that are linked to  $ct_j$   
 $score(link(cs_i, ct_j)) = score(cs_i, ct_j) / |CS_j| + score(link(cs_i, ct_j)) / |CT_i|$   
For every SL chunk,  $cs_i$   
determine set  $Li$  of links originating from  $cs_i$   
 $finalink(cs_i) = max(sort(L_i))$ .

So, for example, the bottom SLA chunk in Figure 2 containing the phrase *the Undo command* has lexical links (determined statistically, see Section 4.2) to four TLA chunks. It prefers the link to the objectival chunk *des Befehls Rückgängig* because that chunk has no links back to other SLA chunks. The fact that the SLA is linked to many TLA chunks weakens the link but it is still strong enough to survive the link-decision process (see Figure 2b). This scheme ranks all the possible chunk links on the basis of word-level lexical correspondences in the SLA→TLA direction and the *one-to-one-ness* of the chunk links bi-directionally. If two adjacent chunks in one language are linked to one chunk in the other language, the system will merge them, asking the user for a new syntactic function. When we ran this automatic linking procedure on the sentence data, we found that its chunking and linking patterns were 90% similar to those done by handcoding during the bootstrapping process.

### 3.2 Encoding off-line knowledge: what abstraction level?

The level of abstraction at which the similarity of two sentences is assessed crucially determines what differences can be perceived between the SL' and SL and the nature of linking will determine what kind of structure changing operations or adaptations can be applied to both SLA and TLA. The SL and TL may diverge for linguistic reasons, as characterised in [Dorr, 1993], which would seem to warrant a full syntactic and semantic level analysis. However, this fails to explain how all TL sentences are derived from their corresponding SL even in a non-complex domain such as that of software manual translation. See [Collins et al., 1996] for further discussion, and for a justification of the abstraction level we currently use. Our translation function is a series of sub-sentential links between SLA and TLA "chunks" at a flattened syntactic dependency level. We weight these links depending on how similar the SLA and TLA chunks are in terms of content and functionality within the sentence. This characterises our translation function for each case.

### 3.3 Weighting the links with adaptability scores

Once the structure of links has been established it then remains to allocate an adaptability score to each link. The basic assumption is the following:

- (i) If an SLA chunk links to a TLA chunk of a different syntactic function, then adapting the TLA to suit some SLA' will be unsafe and should be penalised.
- (ii) The penalty is proportional to the relative distance of the SLA and TLA syntactic functions on the "accessibility hierarchy"<sup>2</sup>

<sup>2</sup> The accessibility hierarchy is a notion motivated by language universal criteria, argued by [Keenan & Comrie, 1977] and we adapt it as a measure for syntactic-functionality distance. The further apart two chunks are on this hierarchy:  $SUBJ \triangleleft OBJ \triangleleft IND - OBJ \triangleleft OBJ - OBJ \triangleleft S - COMP \triangleleft 0 - COMP \triangleleft ADVL$ , with respect to their syntactic labels, the lower the link score.

## 4 Adaptation-Guided Retrieval

As we have shown in the previous sections, the combination of lexical content, one-to-one-ness, and relative syntactic functionality all combine to form our measure of how tenuous a link is between a given SLA and TLA chunk. Note that all three criteria are language independent. The resulting adaptability score for a link lies in the range 0-3 where 3 indicates a highly adaptable link between two chunks which have the same word-content and syntactic functionality in the SLA and TLA. Given our flat description and the one-to-one criteria which our mapping scheme enforces, cases will often contain leftover chunks that were not linked at all, either due to linguistic divergences, novel words in isolation, or other non-linguistic factors as discussed in Section 2 above, so these automatically get a score of zero in both the SLA and TLA. While it may seem that the poor adaptability score then results as a side-effect of our method of case representation, let us point out that our goal here is not to characterise linguistic divergences *per se* but rather to retrieve structures that are manipulable by a data-oriented system without any linguistic knowledge.

Retrieval can be regarded as a process that satisfies the constraints imposed by a memory probe (i.e., the input). Marker passing is a viable technique for retrieval, see [Hendler, 1989], and is used here to gather any cases with a similar structure to the input if their reusability is deemed to be above a certain threshold. Activation passing is appropriate because it propagates simple numeric scores throughout memory. The adaptability scores are integrated into the calculation of activation levels, in other words, the adaptability is coupled with similarity at the chunk level, as discussed in Section 3.1.

**Initiating Activation:** From each word of each input chunk *cii*, issue forth a wave of activation of a fixed level (1.0 say).

**Summing Activation:** When an activation wave of strength *I* reaches a memory chunk *csi* of an example, add *I* to the internal activation of that example. Then add the adaptability score for *csi* to produce the adaptability-based activation.

**Selection Criterion:** When all waves have terminated, collate in a list and sort (in descending order) the internal activation levels of each visited example and return the first *R* examples.

### 4.1 Thresholding 'weeds out' the unadaptable cases

Setting a threshold of adaptability will determine the range of cases that the system is allowed to retrieve for a given input. If the case base is large, the threshold may be set quite high. If a certain input construction is mirrored frequently in the corpus the threshold can be raised even higher, ensuring with some certainty that only adaptable templates will be retrieved. This is good news for translators of highly repetitive text. In the experiments, we set the adaptability level with respect to the SLA' to 3, 2, 1 and 0, allowing only high-level chunk mappings at threshold(3) and all templates at level(0), the latter being equivalent to mere similarity-guided retrieval.

The system's performance at each level of adaptability is evaluated on the basis of the quality of output sentences, that is, the number of mistakes found in the output translations, as judged by a human evaluator (see Figure 3). The 'translation score' is REVERB's prediction regarding translation quality, based on its knowledge about chunk similarity and mappability.

**Retrieval at adaptability threshold: 3** At this highly restrictive level, only those templates are retrieved whose SLA differences are all reconcilable by adapting a chunk of mappability 3.

Threshold	No. of matching pairs (214 cases)	Human evaluation (av. no. mistakes)
0	104	2.57
1	50	1.28
2	8	0.25
3	6	0.00

**Figure 3.** Results of matching REVERB's 214 cases against each other

This usually involves a simple noun-phrase substitution. Coverage is extremely low.

**Result:** 6 matches in 214 cases. **Quality:** perfect translations.

**Retrieval at adaptability threshold: 2** At this level, one or more of the SLA chunks which need adaptation is different in word-form to the TLA, but syntactic functionality is isomorphic. **Result:** 8 matches in 214 cases **Quality:** Same as for threshold 3, but is dependent on the chunk's translation being correct in the lexicon for the context of the new case.

**Retrieval at adaptability threshold: 1** Here, the SLA chunks that need adaptation have different corresponding syntactic functions in the TLA.

**Result:** 50 matches in 214 cases **Quality:** Varies from perfect (no mistakes) in some 36% of the cases to mediocre (3 errors or less) in most others. Here the most common problems are verbal forms and positioning.

**Retrieval at adaptability threshold: 0** One or more chunks in the SLA which need adaptation have no lexically determinable links to the TLA.

**Result:** 104 matches in 214 cases **Quality:** Here, the translation quality suffered from extra TLA' words being present which were irrelevant to the SLA' (e.g. modal and auxiliary verbs). Also, linear order and word-forms of many chunks were incorrect.

## 4.2 Adaptation using the system lexicon

The new target word is inserted into the TLA' structure (a copy of the old TLA example), and assumes the same syntactic function, linear order and syntactic category as the original. If a chunk has been seen before by the system, it will be stored in the system's bilingual lexicon, which is simply a table of word-frames which are linked, in turn, to their host cases and thus to their TL-equivalents within those cases. In other words, no effort goes into building the lexicon as it is a by-product of REVERB's case-memory organisation. Our statistical method of ascertaining word-level translation equivalences is admittedly easily fooled by indirect associations of the type described by [Melamed, 1996]. For example, a given word in an SLA chunk will be associated not only with its corresponding word in the TLA chunk but with all the others as well. We do not see this as a problem however because it simply means that the system produces stronger links for word-groups that have appeared together in the past. It also means

that the system makes equivalences which a "cleaner" lexicon may not have facilitated, as in the *Redo* → *Dieser Befehl* ("this command") translation of Section 3.1.

## 5 Conclusion

In testing our experimental system, REVERB, we found that it can accurately adapt exemplar TLA templates when the corresponding SLA exemplar has sufficient structural similarity to the input sentence. Indeed, the CorelDRAW corpus contains several examples of such near-mappings, which are typical of computer software manuals in general. We see our theory as being of a practical nature and which could be of use in commercial translation domains. Future work will involve scaling up of the system and finer tuned adaptation to the effect that chunks will be adapted chunk-internally. Also, frequently occurring linking patterns in the data could be exploited to generate explicit adaptation rules for a given set of linguistic phenomena.

## References

- [Brown & Frederking, 1995] Brown, Ralf and Frederking, Robert 1995. Applying Statistical English Language Modelling to Symbolic Machine Translation. In *Sixth International Conf. on Theoretical and Methodological Issues in Machine Translation*, Leuven, Belgium, pp. 221 - 239.
- [Collins & Cunningham, 1996] Collins, B., Cunningham, P. 1996. Adaptation Guided Retrieval in EBMT. In Smith, Ian, Faltings, Boi. (eds) *Lecture Notes in Artificial Intelligence 1168*, Springer. pp 91-104.
- [Collins et al., 1996] Collins, B., Cunningham, P., and Veale, T. 1996. An Example Based Approach to Machine Translation. In *Expanding MT Horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas: AMTA-96*, Montreal, Canada, pp. 125-134.
- [Dorr, 1993] Dorr, Bonnie. J. (ed.) 1993. *Machine Translation: A View from the Lexicon*. MIT Press.
- [Hendler, 1989] Hendler, J.A. 1989. Marker Passing over Micro-Features: Toward a Hybrid Symbolic/Connectionist Model, *Cognitive Science 13(1)*.
- [Keenan & Comrie, 1977] Keenan, E.L. and Comrie, B. 1977 Noun Phrase Accessibility and Universal Grammar. *Linguistic Enquiry 8*:63-99.
- [Kolodner, 1993] Kolodner, Janet. 1993. *Case-Based Reasoning*. San Mateo, CA: Morgan Kauffman.
- [Leake et al, 1995] Leake, D, Kinley, A., Wilson, D. 1995 Learning to Improve Case Adaptation by Introspective Reasoning and CBR. In Veloso M., Aamodt A. (eds) *Case-based Reasoning Research and Development: Lecture Notes in AI 1010*. Springer. (1995) pp. 229-240
- [McLean et al., 1994] McLean, Ian, Jones, Daniel, Somers, Harold. 1994 Experiments in Multilingual Example-based Generation, *Proceedings of Cognitive Science of Natural Language Processing 1994*. Dublin, Ireland.
- [Melamed, 1996] Melamed, I Dan (1996) Automatic Construction of Clean Broad-Coverage Translation Lexicons. In *Expanding MT Horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas: AMTA-96* Montreal, Canada, pp. 125 - 134.
- [Streiter & Schmidt-Wigger, 1995] Streiter, Oliver and Schmidt-Wigger, Antje 1995. Patterns of Derivation. In *Expanding MT Horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas: AMTA-96* Montreal, Canada, pp 256 - 272.
- [Sumita et al., 1993] Sumita, E., Oi, K., Iida, H., Higuchi, T., Takahashi, N., Kitano, H. 1993. EBMT on Massively Parallel Processors. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence 1993, IJCAI-93* Chambery, France. pp. 1283 - 1288.
- [Smyth & Keane, 1995] Smyth, Barry, Keane, Mark T. 1995 Experiments on Adaptation-Guided Retrieval in a Case Based Design System. In *M. Veloso & A.Aamodt (eds) CBR Research and Development*, Springer.
- [Voutilainen, 1995] Voutilainen, Atro. 1995 A syntax-based part-of-speech analyser. In *Proceedings of the Seventh Conference of the EACL*, Dublin 1995. pp 157-164.