Gaijin: A Bootstrapping, Template-Driven Approach to Example-Based MT

Tony Veale, Andy Way,

School of Computer Applications, Dublin City University, Dublin, Ireland.

Abstract: Example-based Machine Translation (EBMT) is a recent approach to MT that offers robustness, scalability and graceful degradation, deriving as it does its competence not from explicit linguistic models of source and target languages, but from the wealth of bilingual corpora that are now available. Gaijin is such a system, employing statistical methods, string-matching, case-based reasoning and template-matching to provide a *linguistics-lite* EBMT solution. The only linguistics employed by Gaijin is a psycholinguistic constraint—*the marker hypothesis*—that is minimal, simple to apply, and arguably universal. The scope and current state of Gaijin is described, and some initial evaluation results are reported.

1. Introduction

Example-based, or Memory-based, reasoning is a relatively new paradigm in which scalable machine translation solutions have recently been sought (see Sato & Nagao 1990; Brown *et al.* 1990; Sumita *et al.* 1990; Kaji et al. 1990; Kitano 1993; Somers *et al.* 1994; Brown & Frederking 1995, and Collins *et al.* 1996). One can summarize the spirit of this new EBMT endeavour as a turning away from the traditional Chomskyan emphasis on *competence-modelling* in MT, to emphasise instead the performance aspects of human translation. So rather than attempting to formalize the linguistic competence of native source and target language speakers, EBMT instead exploits the wealth of available performance-data that exists in the form of bilingual text resources from previous human translations. This exploitation usually takes the form of a statistical analysis of bilingual corpora to infer both lexica and translation functions from raw texts, rather than from pre-defined grammars.

As one might expect, EBMT systems are noise-prone statistical engines which do not usually exhibit the syntactic and semantic sharpness of their more linguistically-formalized and knowledge-based MT siblings. However, on the plus side of this comparison lie the issues of robustness, scalability and graceful degradation: if EBMT can be said to produce mediocre results, it manifests this mediocrity in an eminently scalable and robust fashion. Furthermore, even on issues of quality, EBMT is often better placed to maintain the stylistic idiom of a domain when generating target language sentences, due to its statistical grounding in past texts from that domain.

This paper describes the architecture of a example-based MT system named Gaijin, a comprehensive *'soup to nuts'* translation framework. Gaijin is currently being developed in the context of English ↔ German translation, within a textual domain that comprises the help files and user documentation of a popular drawing software package (see also Collins *et al.* 1996). However, Gaijin is conceived for the most part as a linguistics-lite approach to MT, and nothing in its design precludes its application to other languages or text domains. The current paper will concentrate on the bootstrapping character of Gaijin's translation process, describing how example-base creation is initially seeded with a base (possibly empty) set of correspondence statistics, whereupon the examples created are used to later refine these statistics.

2. Gaijin System Architecture

The Gaijin environment consists of the following stages, with the appropriate tools to accomplish each:

- *Bilingual Corpora Alignment*
- *Automatic Lexica construction*
- *Transfer-Template Generation*
- *Example Retrieval*
- *Example Adaptation*
- *New Example Acquisition*

As described in the introduction, Gaijin is a bootstrapped system which initially performs bilingual text alignment on the basis of a seed collection of source:target word correspondences. Though there exists a variety of different text alignment tools to do such a job (e.g., see Gale & Church 1993; Simard *et al.* 1992; Kay & Röscheisen 1993), we feel that the current approach is somewhat less complicated in nature, while making better use via feedback of the statistics that such alignments will eventually allow the system to collect. We describe the above steps, which comprise the system's bootstrapping cycle, in further detail in section 2.7. below.

2.1. Text Alignment

Bilingual sentence alignment can very quickly become a vexing problem of dynamic programming (see Gale & Church 1993). The problem is straightforward when one knows that both source and target texts contain the same number of sentences, as one can make the risky but simplifying assumption that the mapping between both is a planar isomorphism. But this assumption clearly becomes more risky the larger the corpora become. Furthermore, if both texts present an unequal number of sentences, one must try to determine a homomorphic mapping between both that represents a local (if not global) maximum of some measure of alignment suitability. This homomorphism must account for all the possible mis-orderings, mergings and splittings that occur when a human translator does not preserve the sentential structure of the original source.

In an effort to avoid the costs of a dynamic solution, Gaijin exploits the *naïve alignment assumption* that a planar isomorphism will suffice when both texts contain equal numbers of sentences whenever possible. Clearly, the criterion for this assumption is rarely met by most real-world texts. To maximize the applicability of this assumption then, Gaijin first attempts to align both texts at the level of their logical document structure. Such a structure is easy to ascertain the tab and paragraph markings in both texts are extracted to determine where each heading and logical section begins and ends. If two texts contain the same number of headings and sections, they are said to be *structurally-compatible*; otherwise, a naïve alignment of source and target headings is presented to the user to allow him/her to pinpoint exactly where the documents diverge, thus allowing them to be *made* structurally-compatible (often by the simple insertion/deletion of tabs and newlines).

Once both documents are recognizably compatible in logical structure, a planar isomorphism is assumed between the headings and sections of both. This is a relatively safe assumption, for while individual sentences may sometimes be translated out of order, translators rarely do violence to the logical layout of a text. With this structure-mapping in place, corresponding section blocks can then be sentence-aligned, by naïve means if the sentence-number criterion holds, and by dynamic programming driven by sentence length and word-correspondence probabilities when it does not.

2.2. Lexicon Construction

Using a procedure similar to that employed in Kay & Röscheisen (1993) and Somers *et al.* (1994), a correspondence matrix relating the words of the source corpus to those of the target is statistically constructed. A modified variant of *Dice's coefficient* (see van Rijsbergen 1979) is employed to derive a measure of target:source correspondence based both on the absolute frequencies of each source and target word, and upon their frequency of joint occurrence in the same example. Additionally, Gaijin incorporates a mean sentence-length bias into this measure to reflect the different degrees of freedom (i.e., likelihood of error) in each example: the smaller the source and target sentences in a given example relative to the overall corpus mean, the more weighting is given to a source:target correspondence derived from that example; in contrast, examples whose source sentences are longer than the mean are punished, to reflect a the greater likelihood of correspondence error in such examples.

2.3. Example/Template Generation

Unlike the predominantly information-theoretic approach as characterized by the work of Brown *et al.* (1990), Gaijin employs corpus-based statistics not as a translation strategy in themselves, but as a basis for inferring symbolic transfer rules, or translation templates, from bilingual texts. In this respect Gaijin's approach is most akin to the case-based perspective offered by Collins *et al.* (1996). From this perspective, each example in the system's memory serves as a *past case*, a remembered instance of previous linguistic reasoning that can be recalled and adapted to suit current purposes.

A Gaijin template encodes a fixed but variablized mapping between two grammatically-marked sentence structures, one from each of the source and target languages. If a given input source sentence matches the source side of a template, this mapping then provides a hypothetical arrangement of the source sentence's marked elements in the target language. So rather than statistically distil a probabilistic language *distortion model* from all corpus examples taken as a whole, to model how likely each position in a source sentence is to map onto a particular position in the target (see for instance Brown *et al.* 1990), Gaijin instead relies on the availability of each individual example at translation time to potentially provide the correct mapping via template-matching.

2.4. Example/Template Retrieval

A major component of any example-based or case-based engine is the case retrieval mechanism, through which a suitably similar past example is recalled from memory to serve as a translation basis for the current source. A major design criterion for such a recall mechanism is whether the system is to limit its search of memory for a single example to span the current source, or whether multiple smaller examples are to be retrieved, compositionally adapted and then stitched into a final whole. As described in section 5, Gaijin currently pursues a compromise strategy whereby a single template is used to match the input source sentence, but the phrasal segments of the target language that are used to fill this template and provide a translation can be taken from any number of other examples. Planned extensions involve the design of a robust matcher that determines the largest phrasal span of the input that can be covered by a single template, thus allowing the input to be covered in a piecewise fashion.

2.5. Translation Adaptation

Once a matching template has been found for the source input, thus providing a proposed target reorganization of source phrases, it becomes necessary to translate each of these phrasal elements into the target language. Since the template reorganization is not a general one, but one specific to this particular translation example, phrase translation should be done in a manner in keeping with the original human translation. If possible then, the existing translation from the original template should be used if it is applicable, but this situation only arises when the input source phrase is identical to the original example phrase. It is more frequently possible to adapt the original example phrase if the new source phrase only differs by a few words, especially if those words represent merely paradigmatic changes (e.g., "drawing" → "drawings"). If such word-level *surgery* is not practical, a translation for the source phrase must be sought from another example, again ensuring that the translation matches the current example's translation as closely as possible.

2.6. New Example Acquisition

This is perhaps the simplest stage in the Gaijin process. Once a sentence translation has been offered to the user, he/she may signal their approval by asking that the translation and its source be re-entered into the system's memory-base as a new example. This new example may, in turn, extend the adaptive reach of the system to cover future sentence variations unreachable from the existing example-base.

2.7. The Bootstrapping Cycle

While the text alignment algorithm described in section 2.1. above is rather naïve when compared with others of the literature, such as those of Brown *et al.* (1990) and Church & Gale (1993), it is both efficient and open to iterative refinement. Starting with a seed set of word correspondence weights, the system determines a sentence-by-sentence alignment of both texts, within the context of a broader alignment of logical document structures. Once the correspondence matrix of section 2.3 has been constructed, in turn supporting the creation of segment-aligned templates in section 2.4, this matrix can be subsequently refined on the basis of the aligned text segments that underlie these templates.

For instance, the Gaijin system currently operates with an aligned corpus of 1836 examples, containing sentences whose mean source length is eleven words. In turn, when these examples are templatized, they give rise to a 3451-entry corpus of aligned phrases, whose mean source length is only five words. When successively used as a newer basis for determining the system's correspondence matrix, these phrasal examples present significantly fewer degrees of statistical freedom, thus producing more accurate results. Once a more noise-free matrix is calculated, it can be used to align further bilingual corpora from the same domain with higher levels of accuracy.

3. Corpus Statistics and Lexica Construction

We consider an example base $\boldsymbol{E}$ to be comprised of N examples of the form $Ei = <Si, Ti>, 0 > i \leq N$, where Si and Ti are aligned source and target sentences. Given this arrangement, a square correspondence matrix $\boldsymbol{c}$ relating source and target words is inferred from $\boldsymbol{E}$ using a variant of Dice's coefficient, where *lc* denotes a *length-adjusted occurrence count*, and $|E|$ denotes the mean sentence length of the example-base. The greater $|E_i|$ relative to $|E|$, the lower the adjusted word count is deemed to be.

The matrix $\boldsymbol{c}$ can be viewed functionally as underlying three different functions, the unary $\boldsymbol{c}:S \rightarrow T$ and $\boldsymbol{c}:T \rightarrow S$, and the binary $\boldsymbol{c}:S \times T \rightarrow \Re$. Thus, $\boldsymbol{c}(ws) = wt$ states that the best mapping for the source language word *ws* is the target word *wt*, with $\boldsymbol{c}(wt) = ws$ stating the reciprocal case, while $\boldsymbol{c}(ws, wt) = 0.8$ states that a mapping between *ws* and *wt* is to be viewed with a conviction level of 0.8. We use the term 'conviction' rather than 'probability' here as the contents of any given row or column of $\boldsymbol{c}$ do not necessarily sum to 1.0.

3.1. Collecting Word Paradigms

Given a few simplifying assumptions about word inflectional patterns in the languages of concern (here, English and German), Gaijin's commitment to a specific morphological model can often be reduced to a problem of string-matching. For instance, by assuming that most inflectional change is made to the rear of a word string (an unsafe assumption for some languages, such as Hebrew, but relatively robust in the context of many

European languages), a range of basic source and target word paradigms can be inferred from the corpus.

Consider that if *wi* and *wj* are two same-language character strings that share a significant threshold of leading characters (say, > *66%*), there is weak evidence to support a paradigmatic relation between both. However, when the same criterion also holds for their target correspondences $\boldsymbol{c}(wi)$ and $\boldsymbol{c}(wj)$, this provides enough support for the creation of two new paradigmatic groupings, $\{wi, wj\}$ and $\{\boldsymbol{c}(wi), \boldsymbol{c}(wj)\}$. For example, the groupings {drawing, drawings}, {Zeichnung, Zeichnungen} and {aktive, aktiven} are all inferred in the current software domain.

To account for morphological variation, especially those variations not well represented by the corpus, a string-based smoothing function is applied to the contents of $\boldsymbol{c}$ to ensure that paradigmatic relatives have similar weights. If there exist two target words wt and wt' such *that* $\boldsymbol{c}(ws) = wt$ and $wt' \in paradigm(wt)$, then $\mathbf{c}(ws, wt')$ is set to contain the maximum value of both $\boldsymbol{c}(ws, wt')$ and $\boldsymbol{c}(ws, wt'{*}\ stringmatch(wt, wt'))$, where $0.66 > stringmatch(wt, wt')) \leq 1.0$. Thus, if $\boldsymbol{c}(Printer, Drucker) = 0.9$ and $\boldsymbol{c}(Printer, Druckers) = 0.4$, then $\boldsymbol{c}(Printer, Druckers)$ is set to $0.9{*}(8/9) = 0.8$.

4. Template Generation

A template is the mediating structure that allows input source sentences to be matched against stored examples of previous translations in memory. As such, a template is a symbolic entity that exploits unification to relate the organization of phrasal elements of the source side *Si* of an example *Ei* to the organization of corresponding elements on the target side *Ti*. To be useful, template structures must strike a balance between simplicity in their source representation thus facilitating easy retrieval with the necessary complexity to encode the mapping between source and target sentences.

4.1. Sentence Segmentation and the Marker Hypothesis

If a template is to serve as a mapping function between the phrasal constituents of a given pair of source and target sentences, a segmentation algorithm must be applied to these sentences to derive a basic phrase chunking of each. While one can either apply the *full* machinery of syntax to this problem, or employ a minimal linguistic-lite solution, Gaijin opts for the latter to reduce its dependency on particular languages. The marker hypothesis (see Green 1979) is a putative psycholinguistic constraint on grammatical structure that has previously been exploited by Juola (1995) for segmental purposes in MT. This proposed universal, which is equally convenient for our current purposes, states that all natural languages are marked for grammar at the string level by a closed set (or more precisely, in Gaijin terms, a set of closed sets) of specific lexemes and morphemes. That is to say, a system can achieve a basic phrase-segmentation of an input sentence by exploiting a closed list of known marker words to signal the beginning/end of each segment. Gaijin employs the following marker sets, among others, for English:

Prep = {In, Out, On, With, Under, From, To, ...}

Det = {The, Those, These, An, A, ...}

Quant = {All, Some, Many, Few, ...}

Gaijin is also parameterized in its treatment of English and German to start a new segment whenever one of these marker words is encountered; the segment is labelled with the category of its leading marker word, such as Prep, Art, etc. However, a caveat: a marker word is not accepted as starting a new segment when to do so would leave either the previous or the current segment devoid of non-marker content words. This caveat means that every segment contains at least one content word, sensibly viewing the sequence "*Up in the other* window" as a single segment rather than four.

4.2. Segment Alignment

Having segmented the source and target sentences of an example using the marker hypothesis, it is necessary to align these segments to create a variablized mapping that can be reinstantiated during future translation sessions. Given that both source and target sentences can be viewed as a sequence of successive phrases, this segment alignment problem is in effect then a localized version of the global sentence mapping problem described in section 2.1. And as with the global problem, where logical document structure was exploited to constrain the scope of the search process, grammatical sentence structure as furnished by the marker hypothesis can be exploited to limit the scope of segment alignment.

The best target-segment correspondence for each source-segment is found by comparing each source-segment to every segment of the target, employing both segment length and word correspondence weights (as stored in **c**) as match criteria. Ideally, one expects source-segments to map onto target segments of roughly equal word-length, while also expecting the words of matching segments to yield high correspondence scores in **c**. As an additional measure, segment matches are rewarded if the leading marker of both is of the same category type. Thus, a 5-word source segment beginning with "With" that maps to a 4-word source beginning with "Mit" is highly rewarded.

However, marker-based segmentation is a relatively crude mechanism that gives rise to phrasal chunks of different syntactic complexities. For instance, two contiguous noun-phrases (marked by Det) in the source might correspond to a single marked noun-phrase in the target. An important aspect of segment alignment then is segment-merging recognizing when a segment mapping is not 1-to-1 but m-to-n and representing this mapping accordingly. Gaijin currently employs the following simple merge criteria: if multiple contiguous source segments map onto the same target segment, these segments are merged; if a single source segment maps onto two contiguous target segments, both target segments are merged; but if a single source segment maps onto two or more non-contiguous target sentences, the mapping is considered unusable, and the source segment is subsequently not variablized, but stored instead as a string literal in the template. For instance, in an alignment where both "(of colors displayed)" and "(on your monitor)" are deemed to map onto "(Der auf dem Bildschirm angezeigten Farben)", both are merged to form the larger segment "(of colours displayed on your monitor)". In contrast, since one cannot generalize about a poorly understood mapping, a template that contains a non-variablized element is only ever retrieved if the input sentence specifies *exactly* the same text in the same location.

4.3. Template Representation

By variablizing all well-formed segment mappings between source and target sentences, Gaijin produces a translation template for each segment-aligned example in both Prolog and Lisp formats. Though Gaijin is prototyped in Lisp, the following two templates prove Prolog to be the clearer of both formats:

```
% Displayscontrolsfor colouringthe extrudedsurfaces
% Durch Klickenauf dieses Symbol lassen sich Optionenzum Kolorieren
% der extrudiertenFlaechenanzeigen

template(example14, english, german,
         [s(A, _, a14), s(B, prep, b14), s(C, det, c14)],
         [durch, klicken, auf, t(A, prep, a14), t(B, prep, b14), t(C, det, c14),
          anzeigen]).

chunk(english, german,a14, [displays, controls],
      [dieses, symbol, lassen, sich, optionen]).% A

chunk(english, german,b14, [for, coloring],[zum, kolorieren]).% B

chunk(english, german,c14, [the, extruded, surfaces],
                           [der, extrudierten,flaechen]).% C

%% In the maximumbox specify the maximumamount of trap you want to add.
%% Geben Sie im Feld maximumdie maximaleAnzahl von Ueberlappungenan die Sie
%% hinzufuegenmoechten.

template(example24, english, german,
         [s(A, prep, a24), s(B, det, b24), s(C, prep, c24), s(D, pro, d24),
                                                   s(E, prep, e24)],
         [t(A, _, a24), t(B, det, b24), t(C, prep, c24),
                                          t([D|E], prep, [d24,e24])]).

chunk(english,german, a24,  [in, the, maximum, box, specify],
                             [geben, sie, im, feld, maximum]).% A

chunk(english, german, b24, [the, maximum, amount], [die, maximale, anzahl]).% B

chunk(english, german,c24, [of, trap], [von, ueberlappungen]).% C

chunk(english, german,[d24,e24], [you, want, to, add],
                                   [an, die, sie, hinzufuegen,moechten]).% (D E)
```

The key points regarding this template are as follows. Note firstly how each variablized segment contains a reference to the marker type of that segment. A source sentence can thus only match with this template when it possesses the same segment structure. Secondly, each such segment also contains a reference to the original text of the phrase that produced that segment; the aligned phrases are stored in memory as chunks, and are directly retrievable using this unique segment reference. Thirdly, leading target segment words that do not have a significant statistical presence (as determined via **c**) in the matching source segment are shorn off that segment and placed literally into the template. Thus, in example-14, "auf" does not have a presence in "displays controls" and so is shorn off. Fourthly, and perhaps most importantly, though several source segments are merged in the above examples, this merging is represented in the target side of the template only, via compound variables of the form [A|B]. In each template the source side is actually left as uncomplicated as possible, reflecting a basic (non-merged) segmentation of the example source. This permits a direct lookup of the template during example retrieval based solely on a basic segmentation of the input source if the system had to anticipate what kind of merging and splitting might possibly occur to a source segmentation during templatization in order to retrieve an example, template recall would become a considerably more vexing problem of combinatorial dimensions.

5. Example Recall

Given the lengths to which the template generation process goes, in order to leave the source counterpart of every template as uncomplicated as possible, example retrieval is as a result a very straightforward process. It simply suffices to index each example/template in memory on both the exact phrasal chunks it contains, and on some more general index of grammatical structure.

For instance, our templatized example, "*In the maximum box specify the maximum amount of trap you want to add*", is indexed in memory under the strings "*In the maximum box specify*", "*the maximum amount*", "*of trap*", "*you want*" and *"to add*". Likewise it is indexed under a symbolic gloss *Prep-Det-Prep-Pro-Prep*, which is simply constructed by concatenating the individual marker types of the sentence in the order in which they occur. Any input string which exhibits the same marked segmentation will thus produce the same structure index, and have direct access to this example. For the most part then, example retrieval in Gaijin is *not* a process of intensive memory search, but one of near-direct lookup from a structure-indexed hash-table.

When multiple templates are retrieved for an input source sentence based upon its structure index, these templates are prioritized relative to the number of identical phrasal segments they share with the input. For instance, a template corresponding to the example "(In the maximum box specify) (the exact amount)(of trap) (you want) (to add)" will be favoured over one derived from "(In the minimum box specify)(the minimum amount)(of trap)(you want)(to add)", as the former shares four segments while the latter shares only two.

5.1. Example Reusability

An example-base is only as good as the reusability of the examples it contains. But how does one measure this notion of example reusability, or in other words, the likelihood that future input sentences will have the appropriate content to avail of the past experience already stored in memory? One metric currently used in the design of Gaijin is to measure the level of *indexing redundancy* in the example-base that is, how reachable each example is given the indexing schemes used. For instance, the more times a specific structure index is employed in the example-base, the more likely that future sentences will avail of that index (and thus the examples it subsumes). We can then in, a reduced sense (and one must be careful here), extrapolate from the current structure of memory to predict future coverage in the same domain.

At the time of writing Gaijin contains 1836 English:German single-sentence examples in the software domain. The mean level of structure-index redundancy for these examples is 313.13% simply, this means that each index is used to subsume, on average, more than three different examples. Likewise, the mean level *of phrasal redundancy* is 149.25% again, this means that each distinct marked phrase occurs in 1.5 different examples on average. Looking more closely, we note that 44.8% of all indices are used to index a single example, while 55.2% are used to index an average of five examples each. This suggests that half the example-base is considerably more reusable than the other, prompting of course the expected conclusion: many more examples are still required to add balance to the system.

6. Template-filling Strategies for Adapting Past Examples

Example adaptation is required whenever the input text instantiating a template does not match exactly with the original text of the example. Adaptation in Gaijin is modelled via two broad categories: *high-level grafting*, in which an entire phrasal segment of the target sentence is replaced wholesale with another from a different example, and *keyhole surgery*, where individual words in an existing target segment of an example are replaced or morphologically fine-tuned to suit the current translation task.

6.1. Boundary Friction and External Coherence

A phrasal segment in the input sentence is translated by reference to another example in memory whenever that segment does not correspond exactly to the corresponding source segment of the example, and that example segment cannot be adapted (for reasons explained below). As shown in section 4.3., source:target segment mappings (such as "the maximum amount" → "die maximale Anzahl") are also stored in the example-base after an example has been templatized, so it is a simple matter to retrieve all, or any, possible translations for a given phrasal segment of the input as needed

Of course, when substituting phrasal elements on a wholesale basis like this, one encourages the problem of *boundary friction*, perhaps the most vexing issue of example-based translation (e.g., see Somers *et al.* 1994). Friction arises at the graft points of multiple segments when those segments have been drawn from different translation contexts, representing not only different case and thematic roles but different stylistics and registers as well. Gaijin offers no comprehensive solution to this problem, but attempts to alleviate it by ensuring that any translation that is recalled from memory for a given source segment is as compatible with the previous example translation for that template position as possible. This is achieved rather simply by recalling the previous example translation from memory using the segment reference contained in the template position (see again section 4.3.), and choosing that translation (when multiple options exist) which shares the most words (especially marker words) with the previous translation. The intuition here is that in preserving key agreement-carrying words from the original text, the new translation is more likely to slot comfortably into the template, reducing friction with other segments.

6.2. Keyhole Adaptation

This process of segment substitution can frequently be avoided by instead performing a word-level adaptation on those elements of the original translation that do not gel with the current input segment.

Consider the adaptation of a target-language segment $T\alpha$ corresponding to the source segment $S\alpha$, which in turn underlies a template position that is now instantiated with an input source segment $S\beta$. The problem, which is to adapt $T\alpha$ to form a suitable translation $T\beta$ for $S\beta$, is solved by first characterizing the differences between $S\beta$ and $S\alpha$, and then determining how these differences can be projected into $T\alpha$. These differences are characterized as follows:

$add(S\alpha) = \{w \mid w \in S\beta \wedge w \notin S\alpha\}$ $delete(S\alpha) = \{w \mid w \notin S\beta \wedge w \in S\alpha\}$

Using **c**, any additions and deletions to $S\alpha$ can be projected onto $T\alpha$, but first it is necessary to establish an isomorphic mapping between $add(S\alpha)$ and $delete(S\alpha)$. Since Gaijin has no linguistic conception of heads, modifiers, etc., it is essential that every addition be paired with a deletion if the correct substitutions to $T\alpha$ are to be done. The problem is trivial of course when $|add(S\alpha)| = |delete(S\alpha)| \leq 1$, but when more substitutions are involved, some further problem classification is necessary.

Gaijin thus distinguishes among the following four kinds of adaptation:

Trivial Additions: A word *ws* to be added to $S\alpha$ is already statistically represented by some word $wt \in T\alpha$ such that $\boldsymbol{c}(ws, wt) \geq$ some threshold, e.g., 0.1.

Trivial Deletions: A word *ws* to be deleted from Sα is *not already* statistically represented by some word $wt \in T\alpha$ such that $\boldsymbol{c}(ws, wt) \geq$ some threshold, e.g., 0.1.

Marked Swaps: A word *ws* to be added to $S\alpha$ and word *ws'* to be deleted from Sα are both of the same marker type (e.g., Prep → "with" → "on" → "in" etc.)

Paradigmatic Swaps: A word *ws* to be added to $S\alpha$ and a word *ws'* to be deleted from $S\alpha$ are both in the same paradigm (e.g., "window" and "windows").

For instance, in the segment mapping "*you achieve the best font image quality*" → "*Die beste Druckqualitaet erzielen Sie*" the source words "Font" and "Image" may both be trivially deleted, while the word "Print" ("Druck-") are trivially added. Likewise, a translation of "*the current objects*" is easily adapted from "*the current object*" → "*Das aktive Objekt*" by replacing "Objekt" with "Objekte", since "objects" and "object" are known paradigmatic relatives.

Once trivial additions, trivial deletions, and marked and paradigmatic swaps have been performed, the remaining adaptations may be classified as *unconstrained*, since without explicit knowledge an isomorphism between the add and delete sets cannot be constructed. A target segment is considered adaptable then when after the constrained adaptations are performed, $|add(S\alpha)| = |delete(S\alpha)| \leq 1$. For if each set contains exactly one element the substitution is trivial, as when "Die aktive Fenster" is adapted to "Die aktive Zeichnungen" to serve as a translation for "the active drawings".

6.3. Fine-Tuning and Internal Friction

Word-level substitutions, like segment-level substitution, can introduce friction into the innards of a target segment by violating agreement conditions that exist between the words of the segment, e.g., between a determiner and a head noun, or between an adjective and a head noun. While not offering a complete solution, the ability to determine word paradigms via string-matching and corpus frequency does allow the situation to be alleviated somewhat, reducing if not eliminating post-editing effort.

Consider a word *wt* which is inserted into a target segment $T\alpha$ for adaptation purposes. Say the word "Zeichnungen" is substituted into the phrase "Die aktive Fenster" in the head position. Such an adaptation generates an agreement violation between "Zeichnungen" (plural, "drawings") and "aktive" (singular, "active"), but Gaijin does not possess the necessary linguistic model to determine this. However, since the paradigm of "aktive" is easily determined to be {aktive, aktiven}, and ***c****(drawings, aktiven)* > ***c****(drawings, aktive),* the system can exploit statistical reasoning to fine-tune the segment further, and replace "aktive" with "aktiven". Of course, for fine-tuning to work well, one must be confident of the correspondence orderings provided by **c**, and such confidence derives primarily from the scale of the corpora used to build **c**. However, the bootstrapping cycle described in section 2.7. does at least refine the contents of **c** in such a way at a segmental rather than sentence level that makes such fine-tuning somewhat more reliable at smaller scales.

7. Evaluation and Conclusions

Gaijin is currently at a prototype stage where more development effort both in terms of improved statistical models and larger corpora is clearly required, but where some preliminary evaluation can nonetheless be offered.

A test set of word paradigms was chosen for its representativeness of the current software domain, on the informal basis that these words have particularly high frequency in the corpus: {object, objects}, {key, keys}, {drawing, drawings}, {window, windows}, {font, fonts}, {printer, printers}, {blend, blends}, {extrusion, extrusions} and {active, open, actual, current}, the latter being a paradigm of convenience rather that strict of morphology. As it happens, of a corpus of 1836 examples, 791 examples (over 40% of the example-base) contain at least one of these words. Using these examples as a test basis, all test words in the source sentence of each example were replaced with another word from the same paradigm (e.g., active → open, drawing → drawings) to create 791 new test sentences in the source language. These sentences were then translated (using the original example as a basis) with the results classified into three broad categories: *broadly/completely correct* minor errors of fine-tuning are ignored, as this procedure can only be expected to operate well when the system corpus is considerably enlarged); *unadaptable* the word(s) to be adapted either fall into a non-variablized segment of a template (which requires an exact match), or due to an error in segment alignment, could not be reliably located in the target segment; and broadly incorrect the wrong target word is adapted, or the correct target word is adapted using the wrong target counterpart (due to noise in **c**).

Gaijin's performance on this test is currently somewhat mediocre: 63% of the 791 test sentences fall into the broadly-correct category, while 18% are unadaptable and 19% are incorrectly adapted. Most of this mediocrity is rooted in the correspondence matrix **c**, which due to initial errors of sentence alignment in the corpus, still contains pockets of erroneous mappings. These mappings undermine the segment alignment algorithm, which in turn affects the system's ability to locate and adapt segments reliably. More sophisticated string-matching algorithms also need to be employed; for instance, the word sets {key, keys} and {blend, blends} were responsible for a disproportionate amount of the error, as these words rarely occur in isolation in the target domain (e.g., "blend" frequently maps to "Überblendungsgruppe" in the German text, meaning "blend group"). Some responsibility is also borne by the marker hypothesis and the way it is exploited in Gaijin; further refinement is needed to accurately segment source and target sentences in a manner that produces well-formed constituents at roughly the same grammatical resolution.

Current refinement work focuses on bootstrapping the system with a more proven sentence alignment algorithm, such as that of Gale & Church (1993) or Kay & Röscheisen (1993). Other finesses that should have been, and now will be, tested, include the use of fertility probabilities to predict when and how source words are likely to indulge in m-to-n rather than 1-to-1 mappings with the target language.

References

Brown, P. F., J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. Lafferty, R. L. Mercer & P. S. Roossin. (1990). A Statistical Approach to Machine Translation. *Computational Linguistics* **16**(2)

Brown, R. D. & R. E. Frederking. (1995). Applying Statistical Language Modelling to Symbolic Machine Translation, in *the proceedings of the Sixth International Conference on the Theoretical and Methodological in Machine Translation*, Vol II, 354-372, Leuven, Belgium.

Collins, B., P. Cunningham, & T. Veale. (1996). Adaptation-Guided Retrieval for Example-Based Machine Translation, *in the proceedings of AMTA'96, The 2nd conference of the Association for Machine Translation in the Americas.*

Gale, W. A. & K. W. Church. (1993). A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics* **19**(1).

Green, T. R. G. (1979). The Necessity of Syntax Markers. Two experiments with artificial languages. Journal of Verbal Learning and Behavior, **18**, pp 481-496.

Juola, P. (1995). *Learning to Translate: A Psycholinguistic approach to the induction of grammars and transfer functions.* Unpublished Ph.D thesis, Dept. of Computer Science, University of Boulder, Colorado.

Kaji, H., Y. Kida & Y. Morimoto. (1990). Learning Translation Templates from Bilingual Text, in *the proc. of COLING'92, the 1992 conference on Computational Linguistics.*

Kay, M. & M. Röscheisen. (1993). Text-Translation Alignment. *Computational Linguistics* **19**(1).

Kitano, H. (1993). A Comprehensive and Practical Model of Memory-Based Machine Translation, in *the proceeding. of the 1993 International Joint Conference on Artificial Intelligence.*

Sato, S. and M. Nagao. (1990). Toward Memory-based Translation. *In the proceedings of the International Conference on Computational Linguistics,* COLING-90, Helsinki, Finland, August 1990.

Simard, M., G. F. Foster & P. Isabelle. (1992). Using Cognates to Align Sentences in Bilingual Corpora, in *the proceedings of the 4th International Conference on Theoretical and Methodological issues in Machine Translation.*

Somers, H. I. McLean & D. Jones. (1994). Experiments in Multilingual Example-based Generation. *In the proceedings of the 3rd International Conference on the Cognitive Science of Natural Language Processing.* Dublin, Ireland, 1994.

Sumita, E., H. Iida & H. Kohyama. (1990). Translating With Examples: A new approach to Machine Translation. *In the proceedings of the 3rd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language.*

van Rijsbergen, C. J. (1979). *Information Retrieval.* Butterworths.