Practical Glossing by Prioritised Tiling

Victor Poznanski, Pete Whitelock, Jan IJdens, Steffan Corley
Sharp Laboratories of Europe Ltd.
Oxford Science Park, Oxford, OX4 4GA
United Kingdom

{vp,pete,jan,steffan}@sharp.co.uk

Abstract

We present the design of a practical context-sensitive glosser, incorporating current techniques for lightweight linguistic analysis based on large-scale lexical resources. We outline a general model for ranking the possible translations of the words and expressions that make up a text. This information can be used by a simple resource-bounded algorithm, of complexity $O(n \log n)$ in sentence length, that determines a consistent gloss of best translations. We then describe how the results of the general ranking model may be approximated using a simple heuristic prioritisation scheme. Finally we present a preliminary evaluation of the glosser's performance.

1 Introduction

In a lexicalist MT framework such as Shake-and-Bake (Whitelock, 1994), translation equivalence is defined between collections of (suitably constrained) lexical material in the two languages. Such an approach has been shown to be effective in the description of many types of complex bilingual equivalence. However, the complexity of the associated parsing and generation phases leaves a system of this type some way from commercial exploitation. The parsing phase that is needed to establish adequate constraints on the words is of cubic complexity, while the most general generation algorithm, needed to order the words in the target text, is $O(n^4)$ (Poznanski et al. 1996). In this paper, we show how a novel application domain, *glossing*, can be explored within such a framework, by omitting generation entirely and replacing syntactic parsing by a simple combination of morphological analysis and tagging. The poverty of constraints established in this way, and the consequent inaccuracy in translation, is mitigated by providing a menu of alternatives for each gloss. The gloss is automatically updated in the light of user choices. While the availability of alternatives is generally desirable in automatic translation, it is the limitation to glossing which makes it feasible to manage the consistency maintenance required.

Glossing as a technique for elucidating the grammar and lexis of a second language text is well-known from the linguistics literature. Each morpheme in the object language is provided with its meta-language equivalent aligned beneath it. Such a glosser may be used as a tool for second-language improvement (Nerbonne and Smit, 1996), and thus provide an educational alternative to the passive consumption of a (usually low quality) translation. We envisage the glosser's primary use as a tool for cross-language information gathering, and thus think it best not to display grammatical information. Our glosser improves on the use of printed or even on-line dictionaries in several ways:

- The system performs lemmatisation for the user.
- Lightweight analysis resolves part-of-speech ambiguities in context.
- Multi-word expressions, including discontinuous and variable ones, are detected.
- A degree of consistency between system and user choices is maintained.

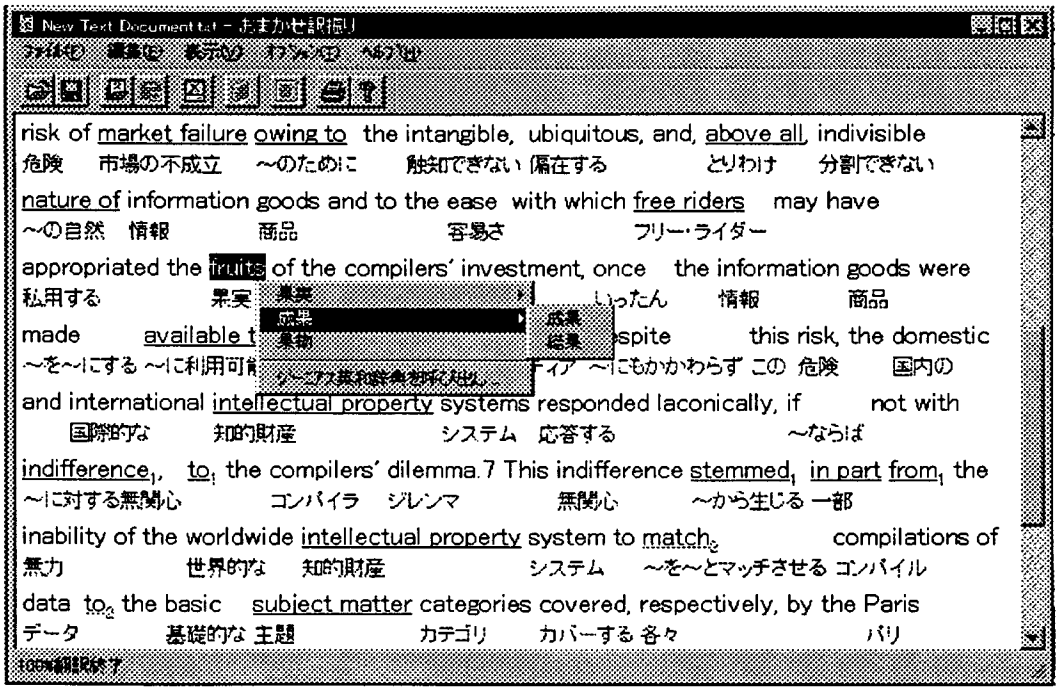

Figure 1: An English to Japanese Gloss

The glosser attempts to find all plausible equivalents for the words and multi-word expressions that constitute a text, displaying the most appropriate consistent subset as its first choice and the remainder within menus. Consistency is maintained by treating source language lexical material as resources that are consumed by the matching of equivalences, so that the latter partially tile the text[1]. Our model has much in common with that of Alshawi (1996), though our linguistic representations are relatively impoverished. Our aim is not true translation but the use of large existing bilingual lexicons for very wide-coverage glossing. We have discovered that the effect of tiling with a large ordered set of detailed equivalences is to provide a close approximation to richer schemes for syntactic analysis.

An example English-Japanese gloss as produced by our system is shown in Figure 1. Multi-word collocations are underlined and discontinuous ones are also given a number (and colour) to facilitate identification. Note how *stemmed ... from* is a discontinuous collocation surrounding the continuous collocation *in part*. The pop-up menu shows the alternatives for *fruit*, by sense at the top-level with run-offs to synonyms, and at the bottom an option to access the machine-readable version of 'Genius', a published English Japanese dictionary.

The structure of this paper is as follows. In 2.1 we outline the basic operation of the system, introducing our representation of natural language collocations as *key descriptors*, and give a probabilistic interpretation for these in 2.2. Section 3 describes the algorithm for tiling a sentence using key descriptors, and goes on to describe a series of heuristics which approximate the full probabilistic model. Section 4 presents the results of a preliminary evaluation of the glosser's performance. Finally in section 5 we give our conclusions and make some suggestions for future improvements to the system.

[1] Equivalences are not only consumers of source language resources but also producers of target language ones. In glossing, the production of target language resources need not be complete – every word needs a translation, but not every word needs a gloss. Tiling thus need only be partial.

2 A Basic Model of a Glosser

To gloss a text, we first segment it into sentences and use the POS tag probabilities assigned by a bigram tagger to order the results of morphological analysis. We obtain a complete tag probability distribution by using the Forwards-Backwards algorithm (see Charniak, 1993) and eliminate only those tags whose probability falls below a certain threshold. Each morphological analysis compatible with one of the remaining tags is passed on to the next phase, together with its associated tag probabilities.

The next phase identifies source words and collocations by matching them against *key descriptors,* which are variable length, possibly discontinuous, word or morpheme *n*-grams. A key descriptor is written:

$W_1\_R_1$ <d_1> $W_2\_R_2$ <d_2> ... <d_{n-1}> $W_n\_R_n$

where $W_i\_R_i$ means a word W_i with morpho-syntactic restrictions R_i, and $W_i\_R_i$ <d_i> $W_{i+1}\_R_{i+1}$ means $W_{i+1}\_R_{i+1}$ must occur within d_i words to the right of $W_i\_R_i$. For example, a key descriptor intended to match the collocation in a fragment like *a procedure used by many researchers for describing the effects* ... might be:

```
procedure_N <5> for_PREP <1> +ing_V0
```

2.1 Collocations and Key Descriptors

We posit the existence of a collocation whenever two or more words or morphemes occur in a fixed syntactic relationship more frequently than would be expected by chance, and which are ideally translated together.

As a linguistic representation of collocations, key descriptors are clearly inadequate. A more correct representation would characterise the stretches spanned by the <d_i> as being of certain categories, or better, that the W_i form a connected piece of dependency representation. However, by:

- expanding the notion of collocation to include a variety of closed-class morphemes,
- refining morpho-syntactic restrictions within the limitations of our current architecture,
- using a very thorough dictionary of such collocations, and
- prioritising key descriptors and using their elements as consumable resources,

we find that the application of key descriptors gives a satisfactory approximation to plausible dependency structures.

Two major carriers of syntactic dependency information in language are category/word-order and closed class elements. Our notion of collocation embraces the full array of closed-class elements that may be associated with a word in a particular dependency structure. This includes governed prepositions and adverbial particles, light verbs, infinitival markers and bound elements such as participial, tense and case affixes. The morphological analysis phase recognises the component structure of complex words and splits them into resources that may be consumed independently.

Those aspects of dependency structure that are not signalled collocationally are often recognisable from particular category sequences and thus can be detected by an *n*-gram tagger. For instance, in English, transitivity is not marked by case or adposition, but by the immediate adjacency of predicate and noun phrase. By distinguishing transitive and intransitive verb tags, we provide further constraints to narrow the range of dependency structures.

2.2 A Probabilistic Characterisation of Collocation

Key descriptors require prioritisation for the tiling phase. In order to effect this, we associate a probabilistic ranking function, f_{kd}, with each key descriptor *kd*.

Consider a collocation such as an English transitive phrasal verb, e.g. *make up*. We may collect all the instances where the component words occur in a sentence in this order with appropriate constraints. By classifying each as a positive or negative instance of this collocation

(in any sense), we can estimate a probability distribution $f_{make\_VT<d>up\_ADV}(d)$ over the number of words, d, separating the elements of this collocation. Suppose then that the tagger has assigned tag probability distributions P^s_{make} and P^s_{up} to the two elements separated by d words in a text fragment, s. The probability that the key descriptor `make_VT <d> up_ADV` correctly matches s is given by:

$$P('make\_VT \langle d \rangle up\_ADV', s) \equiv$$
$$P^s_{make}(VT) \cdot P^s_{up}(ADV) \cdot f_{'make\_VT\langle d \rangle up\_ADV'}(d)$$

More generally,
Eqn (1) :

$$P(kd, s) \equiv \left(\prod_1^n P^s_{w_n}(r_n) \right) \cdot f_{kd}(d_1, d_2, \ldots d_{n-1})$$

where

$$kd \equiv w_1\_r_1 \langle d_1 \rangle w_2\_r_2 \langle d_2 \rangle \ldots \langle d_{n-1} \rangle w_n\_r_n$$

A typical graph of f for the phrasal verb case is depicted in Figure 2. In such cases, we observe that the probability falls slowly over the space of a few words and then sharply at a given d. In other cases, the slope is gentler, but for the vast majority of collocations it decreases monotonically.

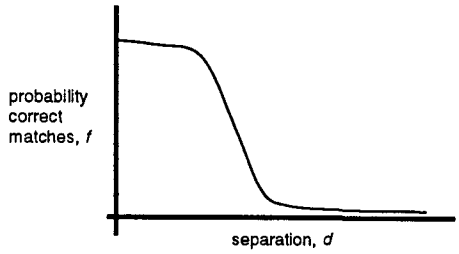

Figure 2: A Typical Frequency Distribution for a Verb Particle Collocation

The overall downward trend in f can be attributed to the interaction of two factors. On the one hand, the total number of true instances follows the distribution of length of phrases that may intervene (in the case of *make up*, noun phrases), i.e. it falls with increasing separation. On the other, the absolute number of false instances remains relatively constant as d varies, and thus increases as a proportion of the total. The fall in true instances is accentuated by the tendency for languages to order dependent phrases with the smallest ones nearest to the head[2], and is thus most marked in the phrasal verb case.

As the number of elements in the equivalence goes up, so does the dimensionality of the frequency distribution. While the multiplied tag probabilities must decrease, the f values increase more , since the corpus evidence tells us that a match comprising more elements is nearly always the correct one.

In section 3.3, we show how we heuristically approximate the various features of f.

3 Glossing as Resource-bounded, Prioritised, Partial Tiling

We prioritise key descriptors to reflect their appropriateness. We then use this ordering to tile the source sentence with a consistent set of key descriptors, and hence their translations. The following sections describe the algorithm.

3.1 General Algorithm

The bilingual equivalences are treated as a simple "one-shot" production system, which annotates a source analysis with all of the possible translations. The tiling algorithm selects the best of these translations by treating bilingual equivalences as *consumers* competing for a *resource* (the right to use a word as part of a translation). In order to make the system efficient, we avoid a global view of linguistic structure. Instead, we assume that every equivalence carries enough information with it to decide whether it has the right to *lock* (claim) a resource. Competing consumers are simply compared in order to decide which has priority. To support this algorithm, it is necessary to associate with every translation a *justification* – the source items from which the target item was derived.

[2] This observation has been extensively explored (in a phrase structure framework) by Hawkins (1994).

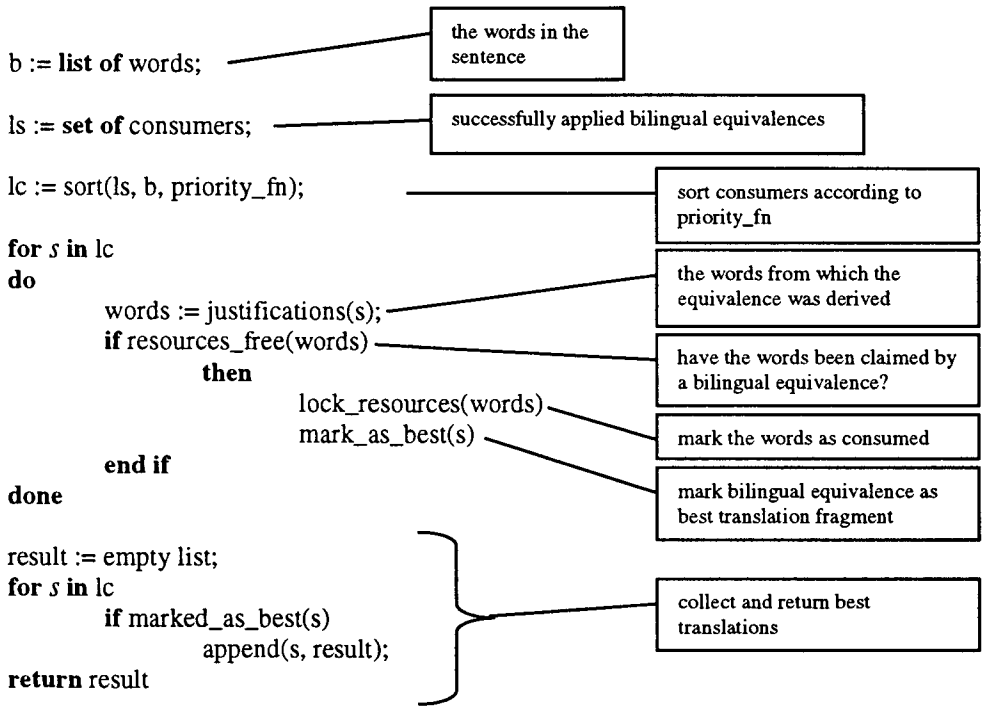

Figure 3: Partial Tiling Algorithm

The algorithm for determining the set of best translations or *translation fringe* is portrayed in Figure 3. The consumers are sorted into priority order and progressively lock the available resources. At the end of this process, the bilingual equivalences that have successfully locked resources comprise the fringe.

3.2 Complexity

We index each bilingual equivalence by choosing the least frequent source word as a key. We retrieve all bilingual equivalences indexed by all the words in a sentence. Retrieval on each key is more or less constant in time. The total number of equivalences retrieved is proportional to the sentence length, n, and their individual applications are constant in time. Thus, the complexity of the rule application phase is order n. The final phase (the algorithm of Figure 3) is fundamentally a sorting algorithm. Since each phase is independent, the overall complexity is bounded to that of sorting, order $n \log n$.

This algorithm does not guarantee to fully tile the input sentence. If full tiling were desired, a tractable solution is to guarantee that every word has at least one bilingual equivalence with a single word key descriptor. However, as will be apparent from Figure 1, glossing the commonest and most ambiguous words would obscure the clarity of the gloss and reduce its precision.

The algorithm as presented operates on source language words in their entirety. Morphological analysis introduces a further complexity by splitting a word into component morphemes, each of which can be considered a resource. The algorithm can be adapted to handle this by ensuring that a key descriptor locks a reading as well as the component morphemes. Once a reading is locked, only morphemes within that reading can be consumed.

3.3 Prioritising Equivalences

If the probabilistic ranking function, f, were elicited by means of corpus evidence, the prioritisation of equivalences would fall out naturally as the solutions to equation 1. In this section, we show how a sequence of simple heuristics can approximate the behaviour of the equation.

We first constrain equivalences to apply only over a limited distance (the search radius),

which we currently assume is the same for all discontinuous key descriptors. This corresponds approximately to the steep fall in the cases illustrated in Figure 2.

After this, we sort the equivalences that have applied according to the following criteria:

1. baggability
2. compactness
3. reading
4. rightmostness
5. frequency priority

Baggability is the number of source words consumed by an equivalence. For instance, in the fragment ... *make up for lost time* ... , we prefer *make up for* (= compensate) over *make up* (= reconcile, apply cosmetics, etc). We indicated in section 2.2 that baggability is generally correct.

However, baggability incorrectly models all values of f in n-dimensional space as higher than any value in $n-1$ dimensional space. In a phrase like *formula milk for crying babies*, baggability will prefer *formula for ... ing* to *formula milk*.

Compactness prefers collocations that span a smaller number of words. Consider the fragment *...get something to eat...* Assume *something to* and *get to* are collocations. The span of *something to* is 2 words and the span of *get to* is 3. Given that their baggability is identical, we prefer the most compact, i.e. the one with the least span. In this case, we correctly prefer *something to*, though we will go wrong in the case of *get someone to eat*. Compactness models the overall downward trend of f.

Reading priority models the tagger probabilities of equation 1. Of course, placing this here in the ordering means that tagger probabilities never override the contribution of f. There are many cases where this is not accurate, but its effect is mitigated by the use of a threshold for tag probabilities – very unlikely readings are pruned and therefore unavailable to the key descriptor matching process.

Reading priority orders equivalences which differ only in the categories they assign to the same words. For instance, in the fragment *the way to London*, the key descriptor `way_N <1> to_PREP` (= road to) will be preferred over `way_N <1> to_TO` (= method of) since the probability of the latter POS for *to* will be lower.

Rightmostness describes how far to the right an expression occurs in the sentence. All other criteria being equal, we prefer the rightmost expression on the grounds that English tends to be right-branching.

Frequency priority picks out a single equivalence from those with the same key descriptor, which is intended to represent its most frequent sense, or at least its most general translation.

4 Evaluation

The above algorithm is implemented in the SID system for glossing English into Japanese[3]. A large dictionary from an existing MT system was used as the basis for our dictionary, which comprises about 200k distinct key descriptors keying about 400k translations. SID reaches a peak glossing speed of about 12,000 words per minute on a 200 MHz Pentium Pro.

To evaluate SID we compared its output with a 1 million word dependency-parsed corpus (based on the Penn TreeBank) and rated as correct any collocation which corresponded to a connected piece of dependency structure with matching tags. We added other correctness criteria to cope with those cases where a collocate is not dependency-connected in our corpus, such as a subject-main verb collocate separated by an auxiliary (*a rally was held*), or a discontinuous adjective phrase (*an interesting man to know*). Correctness is somewhat over-estimated in that a dependent preposition, for example, may not have the intended collocational meaning (it marks an adjunct rather than an argument), but

[3] Available in Japan as part of Sharp's Power E/J translation package on CD-ROM for Windows® 95. A trial version is available for download at http://www.sharp.co.jp/sc/excite/soft_map/ej-a.htm

this appears to be more than offset by tag mismatch cases which might be significant but are not in many particular cases – e.g. *Grand Jury* where *Grand* may be tagged ADJ by SID but NP in Penn, *or passed the bill on to the House*, where *on* may be tagged ADV by SID but IN (= preposition) in Penn.

To obtain a baseline recall figure we ran SID over the corpus with a much lower tag probability threshold and much higher search radius[4], and counted the total number of correct collocations detected anywhere amongst the alternatives.

SID detected a total of c. 150k collocations with its parameters set to their values in the released version[5], of which we judged 110k correct for an overall precision of 72%, which rises to 82% for fringe elements. Overall recall was 98% (75% for the fringe). These figures indicate that the user would have to consult the alternatives for nearly a fifth of collocations (more if we consider sense ambiguities), but would fail to find the right translation in only 2% of cases.

Preliminary inspection of the evaluation results on a collocation by collocation basis reveals large numbers of incorrect key descriptors which could be eliminated, adjusted or further constrained to improve precision with little loss of recall. This leads us to believe that a fringe precision figure of 90% or so might represent the achievable limit of accuracy using our current technology.

5 Conclusion

We have described an efficient and lightweight glossing system that has been used in Sharp products. It is especially useful for quickly "gisting" web and email documents. With a little effort, the user can display the correct translation for the vast majority of the items in a document.

In future work, we hope to approximate more closely the full probabilistic prioritisation model and otherwise improve the key descriptor language, leading to more accurate analysis. We will also explore techniques for extracting collocations from monolingual and bilingual corpora, thereby improving the coverage of the system.

Acknowledgements

We would like to thank our colleagues within Sharp, particularly Simon Berry, Akira Imai, Ian Johnson, Ichiko Sata and Yoji Fukumochi.

References

Alshawi, H. (1996) *Head automata and bilingual tiling: translation with minimal representations.* Proceedings of the 34th ACL, Santa Cruz, California.

Charniak, E. (1993) *Statistical Language Learning.* MIT Press.

Hawkins, John. (1994) *A Performance Theory of Order and Constituency.* Cambridge Studies in Linguistics 73, Cambridge University Press.

Nerbonne, John and Petra Smit (1996) *Glosser-RuG: in Support of Reading.* In Proceedings of 16th COLING, Copenhagen.

Poznanski, V., J.L.Beaven and P. Whitelock (1995) *An Efficient Generation Algorithm for Lexicalist MT.* In Proceedings of the 33rd ACL, MIT.

Whitelock, P.J. (1994) *Shake-and-Bake Translation. In Constraints, Language and Computation.* C.J.Rupp, M.A.Rosner and R.L.Johnson (eds.) Academic Press.

[4] threshold 1%, radius 12

[5] threshold 4%, radius 5