The SLE Example-Based Translation System

Pete Whitelock, Victor Poznanski

Sharp Laboratories of Europe Ltd.
Oxford
{pete,vp}@sharp.co.uk

Abstract

In this paper, we describe a hybrid approach to Machine Translation that exploits a corpus of example translations augmented with resources and techniques from rule-based MT. Our main motivation is to achieve reasonable accuracy for some subdomains with good time and space characteristics. Our architecture is relatively deterministic and therefore quite modest in its consumption of computational resources. At the same time, algorithms inspired by a view of translation in terms of string edits allow us to exploit some of the information available in the corpus to improve accuracy in a way that would be more difficult in other models. We describe the system which we developed at Sharp, illustrate how it exploits syntactic and semantic analysis for improved matching and disambiguation, and analyse our competition results.

1. Introduction

This paper describes the Sharp Laboratories of Europe (SLE) entry to the IWSLT 2006 Evaluation campaign, a Japanese–English translation system for basic travel conversation. Sharp Corporation has pursued research and development in MT for more than 20 years, though almost exclusively in the English to Japanese direction. Aiming for maximally usable results rather than theoretical purity, we have made extensive use of resources that we have accumulated over this period. Nevertheless, our approach does offer some novel perspectives on the field that we think may be of wider interest. These include the interplay of thesaurus and dictionary information in example matching and ambiguity resolution, exploitating the potential of explicit examples.

Our recent work has focused on a relatively lightweight MT system suitable for embedding in a PDA-like device for bi-directional English-Japanese conversation. In this formulaic domain, we view existing translation examples as an invaluable source of large, discontinous, colloquial and often idiosyncratic patterns. Our approach takes as its starting point the work of Nagao (1984), which was loosely [1] characterised as 'translation by analogy', and continues along the lines pursued by Sumita (2003). We think of this line of research as 'edit-based translation'. We determine a source edit transcript (a set of substitutions, insertions and deletions) which transforms the source side of an example in the example base into the input string (the query). We then translate the inputs and outputs of this edit transcript to give a similar transcript for the target language, and apply this target edit transcript to the target side of the example.

The translation of the source items in the edit transcript's input is merely those target language items (words plus positions) with which they are aligned in the example (as determined off-line). The translation of the edit transcript's output is based on a bilingual dictionary and lightweight dependency parse. We analyse the entire query using these resources in a typical rule-based manner, but using the best matching example to assist in disambiguation. We then extract the sub-parts that represent the target edit transcript's output.

We choose the single most similar example as determined by a function of edit distance enriched with semantic similarity. This approach contrasts with combinatorially more extravagant approaches such as those found in SMT (Brown et al, 1990), and EBMT (Brown, 1996) where the translation is assembled from fragments.

Figure 1 shows the major module structure of our system.

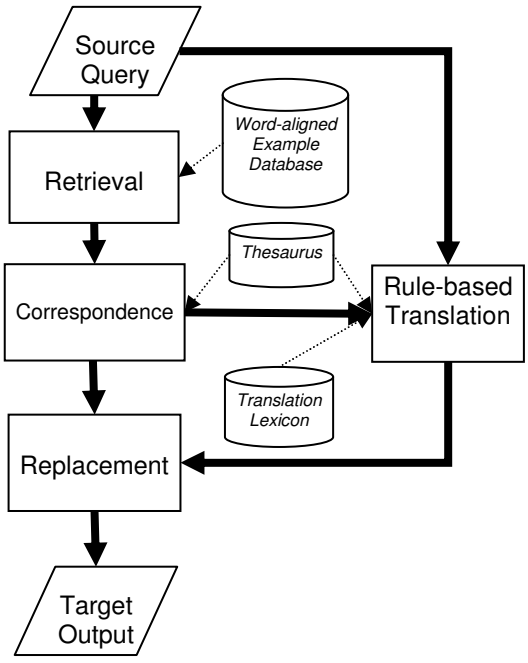

Figure 1: System Architecture

1. As opposed to the pure analogical translation approach of Lepage and Denouval (1995)

The system takes as input a string in the source language, which we call the query, and retrieves a set of candidate examples from the example base. The next stage, correspondance, computes the source edit transcript for the query and each of the candidate examples, and uses this to rank the examples by similarity (Section 2). The query is also analysed by the rule-based translation system, and the best matching example may be used to help resolve ambiguities at this stage (Section 3). The target side of the best matching example (the basis) and the translation of the query are passed to the replacement phase which computes the exact scope of the target edit transcript and applies it to the basis (Section 4).

In the remainder of this paper, we provide a more detailed description of each of the modules, and conclude with a description and discussion of our competition results.

2. Correspondence

The source edit transcript is determined in a stage we call correspondence, which performs an ordered alignment of two strings in the source language. Candidate strings for correspondence with the query are retrieved from the example base using standard vector-space retrieval techniques (Rijsbergen, 1979). Correspondence computes an sequence of alternating matched and unmatched stretches and determines the score based on the lengths of these. Each unmatched stretch comprises the example side (UXS), which is the input of the source edit, and the query side (UQS), which is the output of the source edit. An edit transcript is thus a set of operations of the form UXS => UQS. An empty UXS represents an insertion, an empty UQS a deletion.

As typical of EBMT systems, the score also includes a component for semantic similarity based on a tree-structured thesaurus. A word may be associated with one or more semantic codes; the more similar the codes of two words, the less the cost of substituting one for the other.

For example, given the input:

この 階　に 喫茶店　　が あり ます か。(1)
this floor NI coffee shop GA be　POL Q

and the two stored examples

この階にレストランがありますか。 (2)
Is there a restaurant on this floor?

この階に子供服がありますか。 (3)
Is this the floor for children's clothes?

the first of these will be preferred due to the semantic proximity of 喫茶店 (coffee shop) and レストラン (restaurant), giving the result *Is there a coffee shop on this floor?* On the other hand, given an input such as:

この階にコートがありますか。 (4)

which differs from the previous input by a single word コート (coat or court), we will prefer the second, giving the output *Is this the floor for coats?* (We'll see below how the translation *coat* gets chosen and inflected.)

3. Rule-Based Translation

We have explored various ways to implement the translation of the unmatched query stretches. For instance, a version of the system which appears as part of Sharp's Power EJ Translation Package uses the aligned example base itself. However, this strategy requires a much larger example base, and in the competition version of the system, the translation is based on a separate bilingual dictionary. The dictionary we use is collected from a variety of sources, most of which are ultimately hand-coded, though we can exploit translation frequencies derived automatically by application of the dictionary to the alignment of our example base.

We use a lightweight dependency parser to analyse the input; the dictionary entries may refer to any combination of dependency structure and linear order of items. Bilingual dictionary entries are also labelled with thesaurus codes.

We determine all dictionary entries that could apply to any part of the query. The reason why we don't restrict ourselves to consideration of the unmatched stretches (US) only is that a single dictionary entry may be used to translate material that straddles the matched/unmatched boundary. Since we key dictionary entries by the single least frequent item, and since an entry may contain a variable, the key of an entry that uses material within the US may lie outside the US. In effect we need to expand the US to include anything that is cotranslated with it. For instance, given the input and example:

彼　は　３時　　に　戻り　ます。 (5)
He　TOP 3 o'clock NI return POL

彼　は　月曜日　に　戻り　ます。 (6)
He'll be back on Monday

We need to recognise that the input will use the dictionary entry:

+clocktime に　at_PREP +clocktime (7)

and expand the unmatched stretches to include the particle に, thereby getting the correct translation (8) rather than (9):

He'll be back at 3 o'clock. (8)
*He'll be back on 3 o'clock. (9)

We thus compute a subset of the lexical entries according to a prioritised tiling scheme as used in Poznanski et al. (1998). Entries covering more source language items take precedence. Translation frequency can be used as a tie breaker. Unlike the case of trying to determine the correct lexical entry in isolation, the existence of a matching example can assist in the event of semantic ambiguity. For instance, in (4) above, we can prefer the translation *coat* for the ambiguous コート because our thesaurus tells us that a coat is more like

children's clothes than a (tennis) court is (and also, bearing in mind the two similar examples (2) and (3), a coat is more like children's clothes than a (tennis) court is like a restaurant).

To complete the operation of the translation module, we could combine the target sides of the prioritised lexical entries, mirroring the dependency structure of the source, then linearise the target structure and extract the translation of the unmatched stretches. In fact, as the susbsequent phase may adjust the exact scope of the unmatched stretches, we defer even the combination of lexical entries until after this phase.

4. Replacement

In this phase, we apply the target language edit transcript that we have computed, replacing the target items aligned with the UXS by the translations of the UQS. Our example base is word-for-word aligned off-line using our dictionaries. Incidentally, this allows us to largely determine the senses of ambiguous words used in the examples. The alignment is typically not total – if the unmatched stretch is not aligned, we can fail the plan based on this example and use the next highest scoring example.

This module also makes use of the lightweight dependency parse, allowing us to determine the head or heads within any stretch of words (in either language) – the internal head, and what that stretch is attached to in the remainder of the sentence – the external head.

Deletions from the example are the easiest edits to deal with. The alignment of the deleted material is deleted from the target side of the example (the basis). If the deletion is of the head of a noun phrase, then the associated grammatical elements such as preceding determiners and prepositions are also deleted.

Substitutions may be more complex. For each UXS, we find in the basis the image under alignment of all items in the UXS. These may be discontinuous in the basis, but if they are separated by common words only, the stretches are merged. If multiple stretches remain, we ascertain the head of each, compute the inverse alignment to the heads of the UXS in the source side, and try to find the corresponding items in the input (or query) unmatched stretch (UQS).

For instance when the sentence:

```
[明日       フットボール] の 試合 が 当地
 tomorrow football      NO game GA here
で あり ます か。                          (10)
DE be   POL  Q
```

matches the example:

[今夜$_1$野球$_2$]の試合が当地でありますか。　(11)
Will there be a baseball$_2$ game here tonight$_1$?

the unmatched example stretch (indicated within []) aligns to discontinuous stretches in the target (as shown by co-subscripting). Using semantic proximity we can detect the (sub-) correspondence between 明日 (tomorrow) and 今夜 (tonight), and position the translations correctly, giving:

Will there be a football game here tomorrow? (12)

If no semantically similar elements are discovered, we can use syntactic similarity as a fallback strategy for stretch splitting.

Finally, insertions in the edit are most problematic. This is because we don't know where to position the translation of the UQS. We treat insertions in two different ways, depending on whether the inserted material is adverbial (renyou) or adnominal (rentai). Adverbial insertions are again divided into two cases. Interjections, topics and similar are positioned at the start of the basis, other adverbials at the end. In the case of adnominal insertions, their external head is pulled into the US, turning the insertion into a substitution and giving us a position for the translated material.

In fact, the strategy of pulling the external head into a US is used to solve another problem. Japanese is uniformly head-final, while English noun phrases have mixed headedness (*an open door*, but *a door open to all*). Substitution of adnominals can lead to problems such as the following:

```
次の 電車  は  この ホーム  で  合って
いますか。                              (13)
next train TOP this platform  DE be right
```

ロンドン行きの電車はこのホーム合っていますか。
Is this the right platform for the train to London? (14)

Given the input (13) and stored example (14), we obtain the translation (15)

*Is this the right platform for the train next? (15)

Pulling the head noun into the US means we retranslate it together with its modifier and allows the resulting translation to be ordered correctly by the rules of English. Even if the head is ambiguous, retranslation should not be a problem as this word will have the same semantic code in query and example source which will lead us to choose the same translation.

Having now determined the exact scope of each edit operation, we compute the target string to be inserted/substituted in the basis. The target sides of the lexical entries which apply to any item in the (possibly expanded) UQS are combined by mirroring the dependency structure of the source, then linearised according to an English generation grammar. As they are put into position in the target side of the example, various steps are taken to 'paper over the cracks'[2]. These include treatment of the a/an alternation, removal of multiple prepositions, determiners before pronouns and so on, generation of inflected comparatives and superlatives, etc.

One particularly interesting aspect of this final rendering phase is the copying of features from what is being replaced

2. The 'boundary friction' of Nirenburg et al. (1993)

to what is replacing it. So for instance if the stretch being replaced is headed by a noun, but the replacement has been translated with a verb as head, we nominalise the verb using monolingual information about English. For instance, given the input (16) and example pair (17):

この 電車 は 定刻に 出発 の 予定 です か。 (16)
this train TOP on-time arrival NO plan be Q

この便は定刻に到着の予定ですか。 (17)
Will this flight arrive on time

we obtain a plan for the translation that can be represented as:

Will this t(電車) t(出発) on time? (18)

出発 (departure) is unambiguously a noun in the Japanese sentence (it's followed by the post-nominal particle の). We recognise that this translation is being substituted for a verb in the basis and obtain the verbal equivalent for *departure* from a monolingual dictionary, giving:

Will this train depart on time? (19)

As well as major category changes, we can also copy syntactic features, which is how we achieve the translation coats in *Is this the floor for coats?* discussed above.

This can be contrasted with a standard SMT system where these part-of-speech alternations are built into the translation model and the target language model will settle on the correct one. But because such a translation model contains the cross-product of lexical and part-of-speech alternations, obviously its size and the time to search the space increases much faster than the number of monolingual rules required in our system.

5. Results

Our EBMT system can translate an input only if its example base contains an example which matches sufficiently closely. We use another system to translate when this is not the case. Although development is under way to use the EBMT system's own rule-based translation system, for the competition submission we used a completely independent system, which we will call the Black Box System (BBS).

Table 1 shows the results for our system on various test sets in different configurations. We give the results for the BBS in the first column, then three columns for each of two example base configuration: our own example base (SLE) of 11,913 examples (175,000 Japanese characters, 380,000 words of English, of similar content to the training set provided for the competition); and this example base combined with the competition training set, giving a total of 56,531 examples (1.7m Japanese characters, 1.93m words of English). The column headed 'EBMT Only' gives the scores for the subset of the input that the EBMT system attempted to translate, a percentage of the total input given in the next column. The final column gives the results using the BBS to translate those sentences for which our system failed to find a similar enough example.

The results fall into two classes with regard to quality, with the results on devset2 (IWSLT 2004) and devset3 (IWSLT 2005) massively better than those for devset4 and the test set (IWSLT 2006). This difference may be due partly to the number of reference translations (16 vs. 7), and partly due to overall difficulty. The latter results are intermediate amongst the participants, but the results for eg devset3 are better than any of those achieved in the actual 2005 competition (Eck and Hori 2005). We attribute such results to two factors not found in state-of-the-art SMT (in 2005). The first is the use of examples, which effectively act as large discontinuous elements in a translation model. However, recent work in SMT has started to address this issue directly, eg Chiang, (2005). The second is the potential for the target language stretches which will be replaced to influence the translation which will replace them, a causal interaction with no counterpart in SMT.

	BBS	SLE Example Base			SLE+IWSLT Example Base		
		EBMT Only	%age	+BBS	EBMT Only	%age	+BBS
devset2 (IWSLT2004)	.3524 [7.7607]	.4910 [7.6240]	70.5	.4063 [8.2176]	.5610 [8.927]	75.3	.4663 [8.8784]
devset3 (IWSLT2005)	.3137 [7.5425]	.4994 [7.8347]	66.0	.3930 [8.1415]	.5450 [8.1934]	72.7	.4411 [8.5965]
devset4 (IWSLT2006)	.1917 [5.5127]	.1537 [2.1997]	38.7	.1828 [5.5208]	.1313 [1.4768]	74.2	.1835 [5.6189]
test (asr 1best)						59.4	.1599 [5.3393]
test (correct)	.1797 [5.4599]					60.6	.1726 [5.6497]

Table 1. BLEU [NIST] scores for Rule-Based System (RBS) and EBMT with two example base configurations

Our system configuration is suitable for lower powered machinery with smaller memory. The total data size is under 10Mb for the small example base, under 17Mb for the larger one, including lexicon of 100,000 entries. Translation speed is around 1 second per sentence on a 500 MHz processor with 128 Mb RAM.

6. Acknowledgements

We'd like to thank our colleagues in Japan, particularly Ichiko Sata and Chikashi Nobata, and those at SLE, particularly Michio Wise who implemented most of the system. Thanks also to Phil Edmonds and the anonymous reviewers for valuable comments on earlier versions of the paper.

7. References

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., Roossin, P. S. A statistical approach to machine translation. Computational Linguistics Volume 16 , Issue 2 (June 1990) MIT Press Cambridge, MA, USA, pp. 79 – 85, 1990.

Brown, R.D. "Example-Based Machine Translation in the Pangloss System", Proceedings of the 16th Coling, Copenhagen, 1996.

Chiang, D. "A hierarchical phrase-based model for statistical Machine Translation" Proceedings of the 43rd Annual Meeting of the ACL, 2005.

Eck, M. and C. Hori "Overview of the IWSLT 2005 Evaluation Campaign", Carnegie-Mellon University, Pittsburgh, 2005.

Lepage, Y and E. Denoual "The purest EBMT system ever built: no variables, no templates, no training, examples, just examples, only examples", in Proceedings of the 2nd Workshop on Example-Based Machine Translation, Phuket 2005.

Nagao, M "A Framework of a Mechanical Translation between Japanese and English by Analogy Principle." In A. Elithorn and R. Banerji (eds.) *Artificial and Human Intelligence*, Amsterdam: North Holland, pp.173-180. 1980.

Nirenburg, S., C. Domashnev and D.J.Grannes "Two approaches to matching in example-based machine translation", in 5th TMI, Kyoto, 1993.

Poznanski, V., P. Whitelock, J. Ijdens, S. Corley "Practical Glossing by Prioritised Tiling", Proceedings of the 17th COLING, Montreal, 1998.

Rijsbergen, C. J. "Information Retrieval", London: Butterworths, 1979.

Sumita, E. "EBMT Using DP-Matching Between Word Sequences" in *Recent Advances in Example-based Machine Translation.* Carl, M. and Way, A. (eds). Kluwer Academic Publishers, Dordrecht, pp. 189 – 209, 2003.