Corpus-Driven Bilingual Lexicon Extraction

Michael Rosner

Department of Computer Science and AI,
University of Malta

Abstract. This paper introduces some key aspects of machine translation in order to situate the role of the bilingual lexicon in transfer-based systems. It then discusses the data-driven approach to extracting bilingual knowledge automatically from bilingual texts, tracing the processes of alignment at different levels of granularity. The paper concludes with some suggestions for future work.

1 Machine Translation

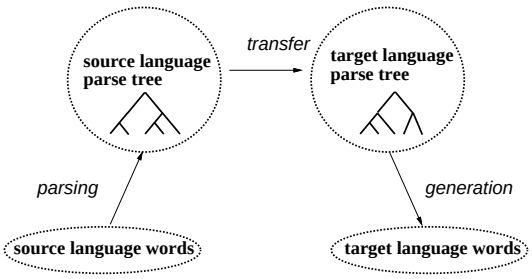

Fig. 1.

The Machine Translation (MT) problem is almost as old as Computer Science, the first efforts at solving it dating from the beginning of the 1950s. At that time, early enthusiasm with the novel and apparently limitless capabilities of digital computers led to grandiose claims that the MT problem would be cracked by purely technological approaches.

However, researchers not only failed to appreciate the computational complexity of the problem but also the role of non-technological factors: syntactic and semantic knowledge, and also subtle cultural conventions that play an important role in distinguishing a good translation from a meaningless one.

By ignoring these factors the systems developed by the early researchers were hopelessly inadequate in terms of both coverage and quality of translation. As a consequence, almost all of the funding for MT dried up in the mid-1960s with a recommendation that more investigation into fundamental linguistic issues and their computational properties was necessary if MT was to move ahead.

This certainly happened. For the next twenty years the emerging field of computational linguistics concentrated upon the development of grammar formalisms – special purpose notations created with the twin goals of (i) encoding linguistic knowledge and (ii) determining computations.

The emphasis on linguistic issues, and on the machinery required to handle them, left its impact upon the proposed architectures for MT systems.

Most Machine Translation (MT) systems are transfer based and figure 1 shows the architecture of a typical transfer-based MT system[1]. The oval items on the left and right represent source and target texts, whilst the arrowed lines denote the three different translation phases.

- **Parsing.** A representation (which need not necessarily be a parse tree, as indicated in the figure) of the surface text is computed, as represented by the left edge of the triangle.
- **Transfer.** The source level representation is transferred to a target representation.
- **Generation.** A surface text in the target langauge is generated from the target level representation.

The essential feature of transfer based systems is that the representation level is not abstract but *linguistic*: a representation from source language S still contains elements (e.g. words) from language S.

A great deal of the character of the system depends upon the *depth* of the representation chosen for transfer. A shallow representation (i.e. one that is close to the surface text), will be relatively easy to compute. From such a representation, it will be relatively easy to generate text. However, precisely because it is shallow, slight variations in the surface form will have to be dealt with by the transfer component. Conversely, a deeper representation, which abstracts away from much of the surface detail, will cause the transfer component to shrink.

Different levels of representation yield a spectrum of different solutions to the MT problem. At one extreme there is Example Based Machine Translation (EBMT) in which there is no representation apart from the text itself. In such systems, the transfer component is essentially lookup (with appropriate indexing) into a database containing pairs of sentences or segment fragments.

At the other extreme are systems like Rosetta (Landsbergen [6]), in which the representation is interlingual (i.e. language free encoding of meaning - see figure 1). Here, the there is no transfer component at all, but the analysis and generation phases are substantial.

1.1 Transfer

The majority of systems fall somewhere in between the two extremes. Systran, for example, which is the technology underlying Babelfish as used by Alta Vista and Google, uses a low level representation (see Wheeler [9]) that includes a limited amount of syntactic information.

The transfer component involves two somewhat different processes.

- Structural transfer deals with the syntactic transformations that are necessary to translate the syntactic conventions of the source language to that of the target. Figure 1.1 shows a simple example that reorders adjectives and nouns (cf. English and Maltese).
- Lexical transfer concerns the actual translations between words. The system component that deals with this kind of equivalence is the bilingual lexicon, as discussed in the next section.

[1] Figure due to Jurafsky and Martin [5]

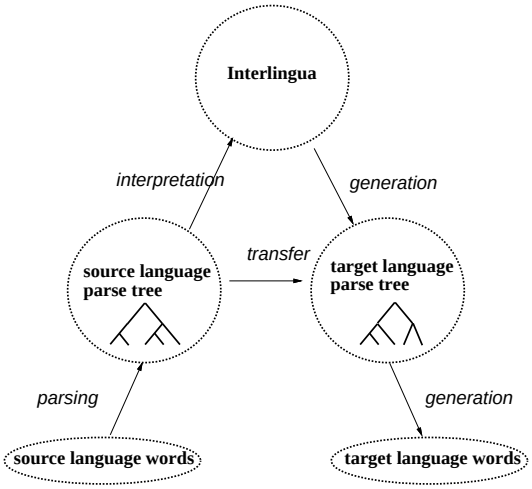

Fig. 2.

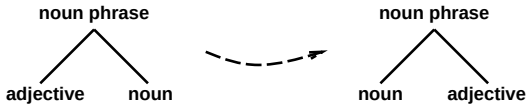

Fig. 3.

1.2 The Bilingual Lexicon

A key element of any translation system is a bilingual lexicon, which, as its name suggests, is essentially a lookup table which encodes mappings between words in the two languages under consideration. The construction of such a lexicon is both non-trivial and highly labour-intensive. The non-triviality arises from several sources. First is the inherently difficult nature of the equivalence relation between words in different languages: for a given word there may be zero to N different possible translations - and where N is greater than 1, the choice may depend upon complex aspects of the context. Second is that the number of words is large: a typical dictionary contains c. 100,000 entries. Even if we ignore the problem of multiple-word expressions and morphological complexity, the creation of such a lexicon is a prodigious task: if each entry takes 15 mins to create (probably an underestimate) this corresponds to 12.5 man-years of work.

In short, then, the creation of a bilingual dictionary constitutes a knowledge acquisition bottleneck, and computational linguists have turned to data-driven rather than handcrafted approaches to lexicon construction.

2 The Data Driven Approach

Professional translators know that the richest source of translation knowledge are texts or text fragments that have already been translated into another language. The question is whether this intuition can be harnessed by an automatic procedure for extracting translation knowledge in general, and a bilingual lexicon in particular, from particular instances of translated texts.

In order to develop this idea, we will regard a text S as a set of *segments*. Segments are simply linguistic entities of a certain type, where the types vary according to the level of analysis. Thus at a given level of analysis, a segment might be a sequence of words, a sequence of paragraphs, or an entire chapter. Furthermore, other relations might be defined over a set of such segments. Within a sentence text, for example, sequence-of-word segments will be ordered. However, for the present we will not be concerned with either the internal character of segments nor with the relations that might exist over them. The important intuition is that the text is a *set*.

3 Alignment

Within the framework under discussion, bilingual knowledge is derived from *correspondences* between a text S and its translation T, that is, between the respective segments in $S = \{s_1, s_2, \ldots, s_n\}$ and $T = \{t_1, t_2, \ldots, t_m\}$. These correspondences are expressed using the notion of *alignment*, defined by Isabelle and Simard [4] as a subset of the Cartesian product $S \times T$.

To give an example, suppose

$$S = \{s_1, s_2, s_3, s_4, s_5\}$$

and

$$T = \{t_1, t_2, t_3, t_4, t_5\}$$

A particular alignment might be

$$A = \{(s_1, t_1), (s_2, t_2), (s_2, t_3), (s_3, t_4), (s_4, t_5), (s_5, t_5)\}$$

This associates the segment s_1 with segment t_1; the segment s_2 with segments t_2 and t_3; the segment s_3 to segment t_4; and the segments s_4 and s_5 to the same segment t_5.

A bilingual text, or "bitext" is a triple (A,S,T) where A is an alignment, S is a text, and T is its translation.

Input to an alignment system is normally the bitext ({(s,t)},{s},{t}), i.e. comprising a single, large text segment, its translation , and the alignment that is the one-to-one mapping between the two.

In order to derive bilingual lexical information from this initial input, a bitext at the next level must be constructed by

- identifying the subsegments of S and T
- computing the best alignment of the subsegments of S and T.

This process is repeated until subsegmentation reaches the word level where, potentially every subset of words in the source text can be aligned with every subset of words in the target text.

At this point we inspect the best alignment and count occurrences of similar pairs. The highest ranking pairs are frequently occurring n-gram-to-n-gram translations and therefore good candidates for the bilingual lexicon.

Of course, this is a deliberate simplification of the process. When, at a given level, the number of subsegments is large, the cost of computing the possible alignments of all possible subsegments is prohibitively high so in practice certain heuristic assumptions must be incorporated to reduce this cost, e.g.

– Subsegments are defined in such a way that that the number of them at a given level is reasonable. This is achieved by staging the alignment process to deal, in order, with alignments of sections, paragraphs, sentences, and finally words.
– Limits are imposed on the set of candidate alignments that will actually be considered. When performing subsentence alignment, for example, one can put an upper bound on their length. Hence, most work in this area in fact deals with single words. We shall see other examples below.

3.1 Paragraph Alignment

Typically, segmentation of paragraphs is carried out in two stages: anchor alignment and paragraph alignment, as described originally by Brown et. al. [2]. A slightly simplified procedure was adopted for the Maltese/English work by Bajada [1].

The two texts are inspected and a set of *anchor point types* manually identified. An anchor point is a point that is easily recognised in both source and target texts such as a chapter heading, title, or other mark that is characteristic of the text type under investigation. The identification of anchor points divides the two texts into *sections* containing one or more paragraphs, as shown in figure 4. A table of anchor point types together with their respective translations is compiled and stored in memory (to be used for computing alignment costs).

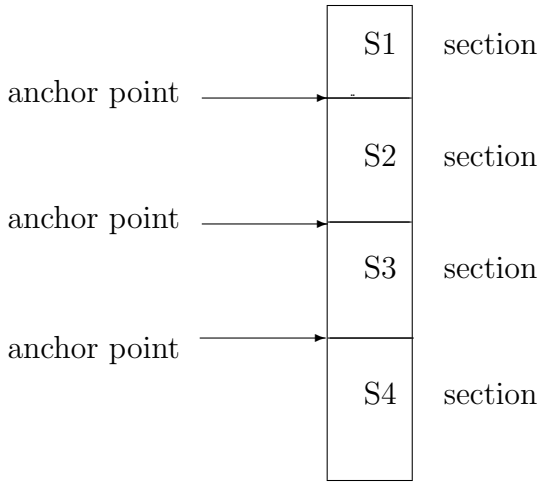

Fig. 4.

For the most part, corresponding anchor points are present in both source and target texts. However, there are typically omissions of one or other anchor and/or minor departures from the exact anchor translations as stored in the table. Hence, the automatic alignment of anchors is not entirely straightforward.

Both Brown et. al. and Bajada adopted the technique of assigning a cost to the alignment of a proposed source/target anchor pairing. If the proposed pair is in the table, the cost is zero. If one element of the pair is an omission, a fixed cost is assigned. Otherwise, rules can be supplied for dealing with concrete cases. In the case of Bajada, a dynamic programming algorithm to find the least cost alignment is based on a limited pattern of possibilities for the *next* anchor pair.

Specifically, two lists of anchor points are created, one for the source and one for the target. The ordering within these list preserves the ordering of anchor points in the texts. The algorithm iterates through each source anchor point for which the first N target anchor points are compared, where N is a small integer, and the one with the least cost assigned. This is an example of limiting the set of candidate alignments that will actually be considered, as mentioned above. In the work carried out by Bajada, a value of 3 was chosen for N, since it was established empirically that for the texts under consideration, the anchor points were never out of alignment by more than 3.

The quality of the results obtained depends largely on the reliability of the anchor points identified by hand as indicators of "section-ness". Although these can be very reliable in formalised text such as legal documents, the problem is harder in the case of, for example, government circulars or communications from the local Water Services. Simard et. al [8] have proposed using *linguistic cognates* as the basis for recognising potential anchor points automatically.

Anchor points provide a rough alignment of source and target sections each containing several unaligned paragraphs.

The next stage aligns the paragraphs within sections. For each aligned anchor pair, the algorithm first retrieves and then counts the number of source and target paragraphs they enclose. Let N and M be the respective counts.

Following the strategy of limiting the set of possible alignments, Bajada allows the following alignment options, for $n, m > 1$:

$$0:0, 1:1, 1:0, 0:1, 1:n, n:1, n:n, n:m(n<m), n:m(m>n), 1:0, 0:1$$

where X:Y means that X source paragraphs are aligned with Y target paragraphs. Clearly, the choice is fully determined for all but the last two cases, for which all possible combinations are enumerated and ranked by considering the length, in characters, of individual paragraphs or concatenations of paragraphs. This strategy proved to be tractable because the number of paragraphs under consideration was always small for the texts being considered.

The underlying assumption is that the length ratio of correctly aligned paragraphs is approximately constant for a given language pair. Hence the cost of a given alignment is proportional to the sum, over all source/target pairs in the alignment, of the difference in character length between each source/target pair, and the least-cost alignment is that which minimises the sum.

3.2 Sentence Alignment

Like paragraph alignment, the sentence alignment algorithms proposed by Brown et. al. [2], and Gale and Church [3], are based on the length of sentences in characters. The latter was reimplemented by Bajada [1]. Like paragraph alignment, both algorithms assume that longer sentences tend to be translated by longer sentences and vice versa. The operation of the algorithm is summarised by Bajada as follows:

> "The algorithm calculates the cost of aligning sentences by assigning a probabilistic score to each possible pair of sentences which make up an alignment. The score is assigned according to the ratio of the lengths of the two sentences and on the variance of this ratio for all sentences. Dynamic programming is then used to find the maximum likelihood alignment of the sentences."

3.3 Word Alignment

In a classic paper that outlines the basic tenets of statistical machine translation, Brown et. al. [7] develop a model in which for estimating Pr(f|e), the probability that f, a target (French) sentence, is the translation of e, a source (English) sentence. This is of course defined in terms of alignments expressed as sets of what they term *connections*.

A connection, written $\mathrm{con}_{j,i}^{f,e}$, states that position j in f is connected to position i in e. The model is directional, so that each position in f is connected to exactly one position in e, including the special "null" position 0 (where it is not connected to any word). Consequently, the number of connections in a given alignment is equal to the length of f.

The translation probability for a pair of sentences is expressed as

$$\Pr(f|e) = \Sigma_{a \in A} \Pr(f, a|e)$$

where A is the set of all possible alignments for the sentences f and e. In other words, each possible alignment contributes its share of probability additively. The question is, how to estimate $\Pr(f, a|e)$. Brown et. al. propose 5 different models.

The first model assumes that $\Pr(f, a|e)$ depends purely on the translation probabilities $t(\phi|\epsilon)$ between the *word* pairs (ϕ, ϵ) contained in the connections of a, and is obtained by multiplying these probabilities together. The individual values for $t(\phi|\epsilon)$ are estimated using a training set comprising a bitext that has been aligned at sentence level.

The second model improves upon the first by taking into account the observation that, under translation, and for certain language pairs at least, words tend to retain their relative position in the sentence: words that are near the beginning of the source sentence tend to appear near the beginning of the target sentence.

3.4 Conclusion

Bajada's FYP report [1] implemented a complete framework for the extraction of bilingual lexical equivalences that included all of the other alignment phases described above, including the two versions of word alignment mentioned above. Using training and test material based on an abridged version of the Malta-EU accession treaty, results for section, paragraph and sentence alignment were encouraging, with precision and recall above 90% over the test material.

The results for word alignment were somewhat less satisfactory, with precision and recall figures closer to 50%. It was shown that the second model for word alignment described above was an improvement on the first.

Amongst the strategies under consideration for improving the performance of the system at word level are:

- Investigation of larger datasets in order to determine the rate of improvement for a given increase in size.
- Use of a lemmatiser to reduce the vocabulary size and hence the frequency of individual connections between source and target words (this will be of particular interest on the Maltese side).
- Investigation of possible equivalences between frequent n-grams of words for given n.

References

1. J. Bajada. Investigation of translation equivalences from parallel texts. *University of Malta, BSc. Final Report*, 2004.
2. P. Brown, J. Lai, and R. Mercer. Aligning sentences in parallel corpora. In *29th Meeting of the Association for Computational Linguistics*, pages 169–176, 1991.
3. W. Gale and K. Church. A program for aligning sentences in bilingual corpora. In *29th Meeting of the Association for Computational Linguistics*, pages 177–184, 1991.
4. Pierre Isabelle and Michel Simard. Propositions pour la reprsentation et lvaluation des alignements et des textes parallles. *Rapport technique du CITI.*, 1996.
5. D. Jurafsky and J. Martin. *Speech and Language Processing*. Prentice-Hall, 2000.
6. J. Landsbergen. Isomorphic grammars and their use in the rosetta translation system. In M. King, editor, *Machine Translation Today*. Edinburgh University Press, 1997.
7. Brown P., J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, and R. Mercer. A statistical approach to language translation. *Computational Linguistics*, 16.2:79–85, 1990.
8. Michel Simard, George F. Foster, and Pierre Isabelle. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research*, pages 1071–1082. IBM Press, 1993.
9. P. Wheeler. Systran. In M. King, editor, *Machine Translation Today*, pages 192–208. Edinburgh University Press, 1997.