13 *Statistical Alignment and Machine Translation*

MACHINE TRANSLATION, the automatic translation of text or speech from one language to another, is one the most important applications of NLP. The dream of building machines that let people from different cultures talk to each other easily is one of the most appealing ideas we as NLP researchers can use to justify our profession (and to get money from funding agencies).

Unfortunately, machine translation (MT) is a hard problem. It is true that nowadays you can buy inexpensive packages that call themselves translation programs. They produce low-quality translations which are sufficient for a translator who can post-edit the output or for people who know enough about a foreign language to be able to decipher the original with the help of a buggy translation. The goal of many NLP researchers is instead to produce close to error-free output that reads fluently in the target language. Existing systems are far from this goal for all but the most restricted domains (like weather reports, Isabelle 1987).

Why is machine translation hard? The best way to answer this question is to look at different approaches to MT that have been pursued. Some important approaches are schematically presented in figure 13.1.

WORD FOR WORD

The simplest approach is to translate *word for word* (the bottom arrow in figure 13.1). One obvious problem with this is that there is no one-to-one correspondence between words in different languages. Lexical ambiguity is one reason. One of the examples we discussed in chapter 7 is the English word *suit* which has different translations in French, depending on whether it means 'lawsuit' or 'set of garments.' One needs to look at context larger than the individual word to choose the correct French translation for ambiguous words like *suit*.

Another challenge for the word-for-word approach is that languages

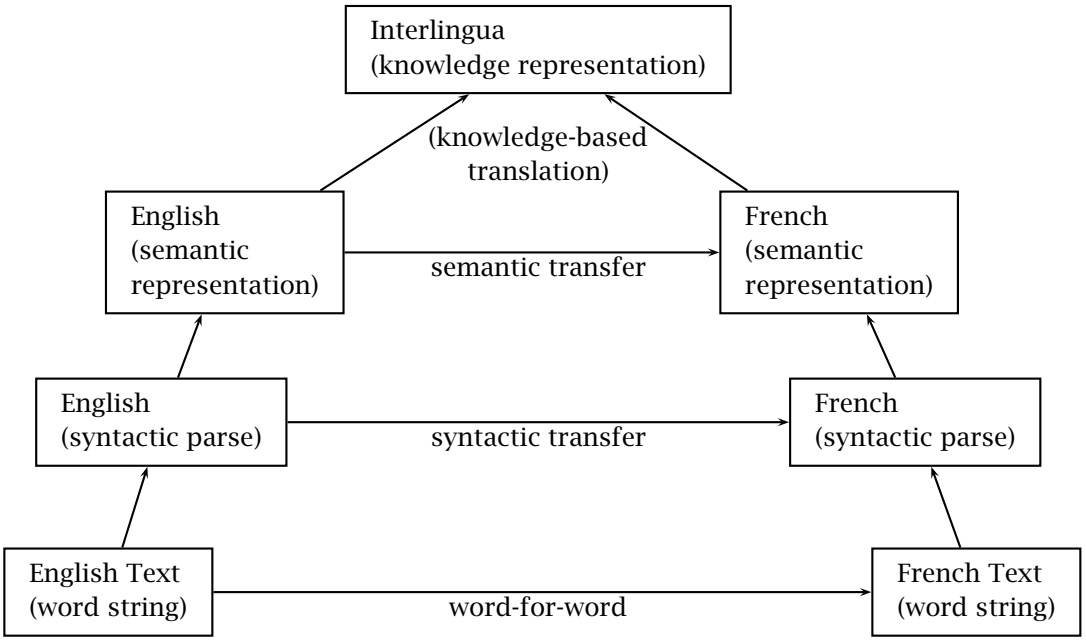

Figure 13.1 Different strategies for Machine Translation. Examples are for the case of translation from English (the source) to French (the target). Word-based methods translate the source word by word. Transfer methods build a structured representation of the source (syntactic or semantic), transform it into a structured representation of the target and generate a target string from this representation. Semantic methods use a richer semantic representation than parse trees with, for example, quantifier scope disambiguated. Interlingua methods translate via a language-independent knowledge representation. Adapted from (Knight 1997: figure 1).

have different word orders. A naive word-for-word translation will usually get the word order in the target language wrong. This problem is addressed by the *syntactic transfer approach*. We first parse the source text, then transform the parse tree of the source text into a syntactic tree in the target language (using appropriate rules), and then generate the translation from this syntactic tree. Note that we are again faced with ambiguity, syntactic ambiguity here, since we are assuming that we can correctly disambiguate the source text.

SYNTACTIC TRANSFER APPROACH

The syntactic transfer approach solves problems of word order, but of-

ten a syntactically correct translation has inappropriate semantics. For example, German *Ich esse gern* 'I like to eat' is a verb-adverb construction that translates literally as *I eat readily* (or *willingly*, *with pleasure*, *gladly*). There is no verb-adverb construction in English that can be used to express the meaning of *I like to eat*. So a syntax-based approach cannot work here.

SEMANTIC TRANSFER APPROACH

In *semantic transfer approaches*, we represent the meaning of the source sentence (presumably derived via an intermediate step of parsing as indicated by the arrows in figure 13.1), and then generate the translation from the meaning. This will fix cases of syntactic mismatch, but even this is not general enough to work for all cases. The reason is that even if the literal meaning of a translation is correct, it can still be unnatural to the point of being unintelligible. A classic example is the way that English and Spanish express direction and manner of motion (Talmy 1985). In Spanish, the direction is expressed using the verb and the manner is expressed with a separate phrase:

(13.1) La botella entró a la cueva flotando.

With some effort, English speakers may be able to understand the literal translation 'the bottle entered the cave floating.' But if there are too many such literal translations in a text, then it becomes cumbersome to read. The correct English translation expresses the manner of motion using the verb and the direction using a preposition:

(13.2) The bottle floated into the cave.

INTERLINGUA

An approach that does not rely on literal translations is translation via an *interlingua*. An interlingua is a knowledge representation formalism that is independent of the way particular languages express meaning. An interlingua has the added advantage that it efficiently addresses the problem of translating for a large number of languages, as is, for example, necessary in the European Community. Instead of building $O(n^2)$ translation systems, for all possible pairs of languages, one only has to build $O(n)$ systems to translate between each language and the interlingua. Despite these advantages, there are significant practical problems with interlingua approaches due to the difficulty of designing efficient and comprehensive knowledge representation formalisms and due to the large amount of ambiguity that has to be resolved to translate from a natural language to a knowledge representation language.

Where do statistical methods come into play in this? In theory, each of the arrows in figure 13.1 can be implemented based on a probabilistic model. For example, we can implement the arrow from the box "English Text (word string)" to the box "English Text (syntactic parse)" as a probabilistic parser (see chapters 11 and 12). Some components not shown in the figure could also be implemented statistically, for example, a word sense disambiguator. Such probabilistic implementation of selected modules is in fact the main use of statistical methods in machine translation at this point. Most systems are a mix of probabilistic and non-probabilistic components. However, there are a few completely statistical translation systems and we will describe one such system in section 13.3.

So why do we need a separate chapter on machine translation then, if a large part of the probabilistic work done for MT, such as probabilistic parsing and word sense disambiguation, is already covered in other chapters? Apart from a few specific MT problems (like probabilistic transfer, see the Further Reading), there is one task that mainly comes up in the MT context, the task of *text alignment*. Text alignment is not part of the translation process per se. Instead, text alignment is mostly used to create lexical resources such as bilingual dictionaries and parallel grammars, which then improve the quality of machine translation.

Surprisingly, there has been more work on text alignment in Statistical NLP than on machine translation proper, partly due the above-mentioned fact that many other components of MT systems like parsers and disambiguators are not MT-specific. For this reason, the bulk of this chapter will be about text alignment. We will then briefly discuss word alignment, the step necessary after text alignment for deriving a bilingual dictionary from a parallel text. The last two sections describe the best known attempt to construct a completely statistical MT system and conclude with some suggestions for further reading.

13.1 Text Alignment

PARALLEL TEXTS
BITEXTS

A variety of work has applied Statistical NLP methods to multilingual texts. Most of this work has involved the use of *parallel texts* or *bitexts* – where the same content is available in several languages, due to document translation. The parallel texts most often used have been parliamentary proceedings (*Hansards*) or other official documents of countries with multiple official languages, such as Canada, Switzerland and Hong

HANSARDS

Kong. One reason for using such texts is that they are easy to obtain in quantity, but we suspect that the nature of these texts has also been helpful to Statistical NLP researchers: the demands of accuracy lead the translators of this sort of material to to use very consistent, literal translations. Other sources have been used (such as articles from newspapers and magazines published in several languages), and yet other sources are easily available (religious and literary works are often freely available in many languages), but these not only do not provide such a large supply of text from a consistent period and genre, but they also tend to involve much less literal translation, and hence good results are harder to come by.

ALIGNMENT

Given that parallel texts are available online, a first task is to perform gross large scale *alignment*, noting which paragraphs or sentences in one language correspond to which paragraphs or sentences in another language. This problem has been well-studied and a number of quite successful methods have been proposed. Once this has been achieved, a second problem is to learn which words tend to be translated by which other words, which one could view as the problem of acquiring a bilingual dictionary from text. In this section we deal with the text alignment problem, while the next section deals with word alignment and induction of bilingual dictionaries from aligned text.

13.1.1 Aligning sentences and paragraphs

Text alignment is an almost obligatory first step for making use of multilingual text corpora. Text alignment can be used not only for the two tasks considered in the following sections (bilingual lexicography and machine translation), but it is also a first step in using multilingual corpora as knowledge sources in other domains, such as for word sense disambiguation, or multilingual information retrieval. Text alignment can also be a useful practical tool for assisting translators. In many situations, such as when dealing with product manuals, documents are regularly revised and then each time translated into various languages. One can reduce the burden on human translators by first aligning the old and revised document to detect changes, then aligning the old document with its translation, and finally splicing in changed sections in the new document into the translation of the old document, so that a translator only has to translate the changed sections.

The reason that text alignment is not trivial is that translators do not al-

ways translate one sentence in the input into one sentence in the output, although, naturally, this is the most common situation. Indeed, it is important at the outset of this chapter to realize the extent to which human translators change and rearrange material so the output text will flow well in the target language, even when they are translating material from quite technical domains. As an example, consider the extract from English and French versions of a document shown in figure 13.2. Although the material in both languages comprises two sentences, note that their content and organization in the two languages differs greatly. Not only is there a great deal of reordering (denoted imperfectly by bracketed groupings and arrows), but large pieces of material can just disappear: for example, the final English words *achieved above-average growth rates*. In the reordered French version, this content is just implied from the fact that we are talking about how in general sales of soft drinks were higher, *in particular, cola drinks*.

In the sentence alignment problem one seeks to say that some group of sentences in one language corresponds in content to some group of sentences in the other language, where either group can be empty so as to allow insertions and deletions. Such a grouping is referred to as a sentence alignment or *bead*. There is a question of how much content has to overlap between sentences in the two languages before the sentences are said to be in an alignment. In work which gives a specific criterion, normally an overlapping word or two is not taken as sufficient, but if a clause overlaps, then the sentences are said to be part of the alignment, no matter how much they otherwise differ. The commonest case of one sentence being translated as one sentence is referred to as a 1:1 sentence alignment. Studies suggest around 90% of alignments are usually of this sort. But sometimes translators break up or join sentences, yielding 1:2 or 2:1, and even 1:3 or 3:1 sentence alignments.

BEAD

Using this framework, each sentence can occur in only one bead. Thus although in figure 13.2 the whole of the first French sentence is translated in the first English sentence, we cannot make this a 1:1 alignment, since much of the second French sentence also occurs in the first English sentence. Thus this is an example of a 2:2 alignment. If we are aligning at the sentence level, whenever translators move part of one sentence into another, we can only describe this by saying that some group of sentences in the source are parallel with some group of sentences in the translation. An additional problem is that in real texts there are a surprising number of cases of *crossing dependencies*, where the order of

CROSSING DEPENDENCIES

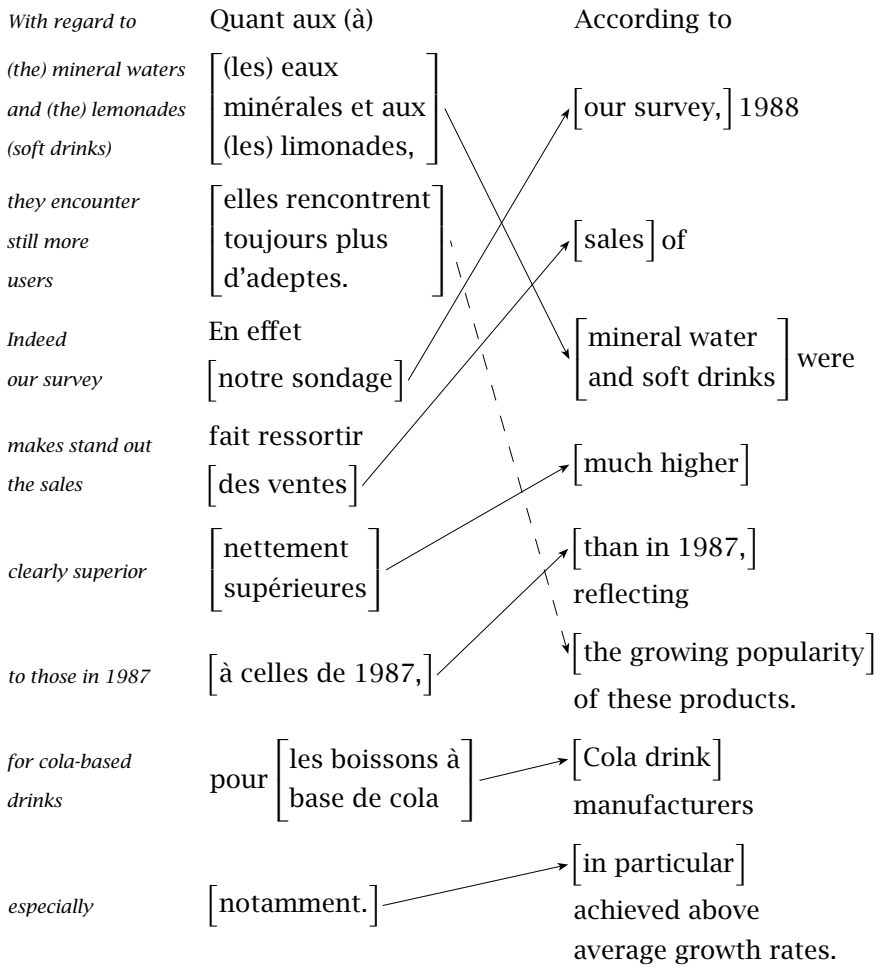

Figure 13.2 Alignment and correspondence. The middle and right columns show the French and English versions with arrows connecting parts that can be viewed as translations of each other. The italicized text in the left column is a fairly literal translation of the French text.

Paper	Languages	Corpus	Basis
Brown et al. (1991c)	English, French	Canadian Hansard	# of words
Gale and Church (1993)	English, French, German	Union Bank of Switzerland reports	# of characters
Wu (1994)	English, Cantonese	Hong Kong Hansard	# of characters
Church (1993)	various	various (incl. Hansard)	4-gram signals
Fung and McKeown (1994)	English, Cantonese	Hong Kong Hansard	lexical signals
Kay and Röscheisen (1993)	English, French, German	Scientific American	lexical (not probabilistic)
Chen (1993)	English, French	Canadian Hansard EEC proceedings	lexical
Haruno and Yamazaki (1996)	English, Japanese	newspaper, magazines	lexical (incl. dictionary)

Table 13.1 Sentence alignment papers. The table lists different techniques for text alignment, including the languages and corpora that were used as a testbed and (in column "Basis") the type of information that the alignment is based on.

sentences are changed in the translation (Dan Melamed, p.c., 1998). The algorithms we present here are not able to handle such cases accurately. Following the statistical string matching literature we can distinguish between *alignment* problems and *correspondence* problems, by adding the restriction that alignment problems do not allow crossing dependencies. If this restriction is added, then any rearrangement in the order of sentences must also be described as a many to many alignment. Given these restrictions, we find cases of 2:2, 2:3, 3:2, and, in theory at least, even more exotic alignment configurations. Finally, either deliberately or by mistake, sentences may be deleted or added during translation, yielding 1:0 and 0:1 alignments.

ALIGNMENT
CORRESPONDENCE

A considerable number of papers have examined aligning sentences in parallel texts between various languages. A selection of papers is shown in table 13.1. In general the methods can be classified along several dimensions. On the one hand there are methods that are simply length-based versus those methods that use lexical (or character string) content. Secondly, there is a contrast between methods that just give an average alignment in terms of what position in one text roughly corresponds with a certain position in the other text and those that align sentences to form

sentence beads. We outline and compare the salient features of some of these methods here. In this discussion let us refer to the parallel texts in the two languages as S and T where each is a succession of sentences, so $S = (s_1, \ldots, s_I)$ and $T = (t_1, \ldots, t_J)$. If there are more than two languages, we reduce the problem to the two language case by doing pairwise alignments. Many of the methods we consider use dynamic programming methods to find the best alignment between the texts, so the reader may wish to review an introduction of dynamic programming such as Cormen et al. (1990: ch. 16).

13.1.2 Length-based methods

Much of the earliest work on sentence alignment used models that just compared the lengths of units of text in the parallel corpora. While it seems strange to ignore the richer information available in the text, it turns out that such an approach can be quite effective, and its efficiency allows rapid alignment of large quantities of text. The rationale of length-based methods is that short sentences will be translated as short sentences and long sentences as long sentences. Length usually is defined as the number of words or the number of characters.

Gale and Church (1993)

Statistical approaches to alignment attempt to find the alignment A with highest probability given the two parallel texts S and T:

$$\arg\max_A P(A|S,T) = \arg\max_A P(A,S,T) \tag{13.3}$$

To estimate the probabilities involved here, most methods decompose the aligned texts into a sequence of aligned beads $(B_1, \ldots, B_K)$, and suggest that the probability of a bead is independent of the probability of other beads, depending only on the sentences in the bead. Then:

$$P(A,S,T) \approx \prod_{k=1}^{K} P(B_k) \tag{13.4}$$

The question then is how to estimate the probability of a certain type of alignment bead (such as 1:1, or 2:1) given the sentences in that bead.

The method of Gale and Church (1991; 1993) depends simply on the length of source and translation sentences measured in characters. The

hypothesis is that longer sentences in one language should correspond to longer sentences in the other language. This seems uncontroversial, and turns out to be sufficient information to do alignment, at least with similar languages and literal translations.

The Union Bank of Switzerland (UBS) corpus used for their experiments provided parallel documents in English, French, and German. The texts in the corpus could be trivially aligned at a paragraph level, because paragraph structure was clearly marked in the corpus, and any confusions at this level were checked and eliminated by hand. For the experiments presented, this first step was important, since Gale and Church (1993) report that leaving it out and simply running the algorithm on whole documents tripled the number of errors. However, they suggest that the need for prior paragraph alignment can be avoided by applying the algorithm they discuss twice: firstly to align paragraphs within the document, and then again to align sentences within paragraphs. Shemtov (1993) develops this idea, producing a variant dynamic programming algorithm that is especially suited to dealing with deletions and insertions at the level of paragraphs instead of just at the sentence level.

Gale and Church's (1993) algorithm uses sentence length to evaluate how likely an alignment of some number of sentences in L_1 is with some number of sentences in L_2. Possible alignments in the study were limited to {1:1, 1:0, 0:1, 2:1, 1:2, 2:2}. This made it possible to easily find the most probable text alignment by using a dynamic programming algorithm, which tries to find the minimum possible distance between the two texts, or in other words, the best possible alignment. Let $D(i, j)$ be the lowest cost alignment between sentences $s_1, \ldots, s_i$ and $t_1, \ldots, t_j$. Then one can recursively define and calculate $D(i, j)$ by using the obvious base cases that $D(0, 0) = 0$, etc., and then defining:

$$D(i,j) = \min \begin{cases} D(i, j-1) + \text{cost}(0{:}1 \text{ align } \varnothing, t_j) \\ D(i-1, j) + \text{cost}(1{:}0 \text{ align } s_i, \varnothing) \\ D(i-1, j-1) + \text{cost}(1{:}1 \text{ align } s_i, t_j) \\ D(i-1, j-2) + \text{cost}(1{:}2 \text{ align } s_i, t_{j-1}, t_j) \\ D(i-2, j-1) + \text{cost}(2{:}1 \text{ align } s_{i-1}, s_i, t_j) \\ D(i-2, j-2) + \text{cost}(2{:}2 \text{ align } s_{i-1}, s_i, t_{j-1}, t_j) \end{cases}$$

For instance, one can start to calculate the cost of aligning two texts as indicated in figure 13.3. Dynamic programming allows one to efficiently consider all possible alignments and find the minimum cost alignment $D(I, J)$. While the dynamic programming algorithm is quadratic,

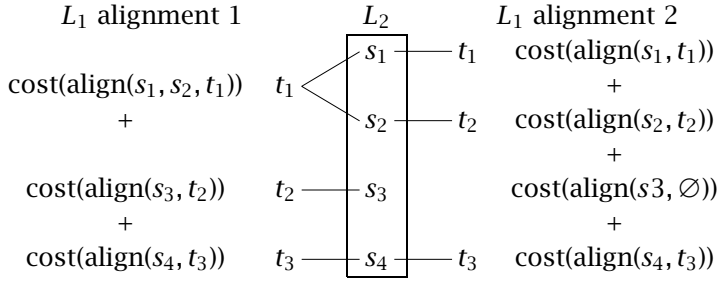

Figure 13.3 Calculating the cost of alignments. The costs of two different alignments are computed, one in the left column (which aligns t_1 with s_1 and s_2 and aligns t_2 with s_3) and one in the right column (which aligns s_3 with the empty sentence).

since it is only run between paragraph anchors, in practice things proceed quickly.

This leaves determining the cost of each type of alignment. This is done based on the length in characters of the sentences of each language in the bead, l_1 and l_2. One assumes that each character in one language gives rise to a random number of characters in the other language. These random variables are assumed to be independent and identically distributed, and the randomness can then be modeled by a normal distribution with mean μ and variance s^2. These parameters are estimated from data about the corpus. For μ, the authors compare the length of the respective texts. German/English = 1.1, and French/English = 1.06, so they are content to model μ as 1. The squares of the differences of the lengths of paragraphs are used to estimate s^2.

The cost above is then determined in terms of a distance measure between a list of sentences in one language and a list in the other. The distance measure δ compares the difference in the sum of the lengths of the sentences in the two lists to the mean and variance of the whole corpus: $\delta = (l_2 - l_1\mu)/\sqrt{l_1 s^2}$. The cost is of the form:

$$\text{cost}(l_1, l_2) = -\log P(\alpha \text{ align} | \delta(l_1, l_2, \mu, s^2))$$

where α align is one of the allowed match types (1:1, 2:1, etc.). The negative log is used just so one can regard this cost as a 'distance' measure: the highest probability alignment will correspond to the shortest 'distance' and one can just add 'distances.' The above probability is calculated using Bayes' law in terms of $P(\alpha \text{ align})P(\delta|\alpha \text{ align})$, and therefore

the first term will cause the program to give a higher a priori probability to 1:1 matches, which are the most common.

So, in essence, we are trying to align beads so that the length of the sentences from the two languages in each bead are as similar as possible. The method performs well (at least on related languages like English, French and German). The basic method has a 4% error rate, and by using a method of detecting dubious alignments Gale and Church are able to produce a best 80% of the corpus on which the error rate is only 0.7%. The method works best on 1:1 alignments, for which there is only a 2% error rate. Error rates are high for the more difficult alignments; in particular the program never gets a 1:0 or 0:1 alignment correct.

Brown et al. (1991c)

The basic approach of Brown et al. (1991c) is similar to Gale and Church, but works by comparing sentence lengths in words rather than characters. Gale and Church (1993) argue that this is not as good because of the greater variance in number of words than number of characters between translations. Among the salient differences between the papers is a difference in goal: Brown et al. did not want to align whole articles, but just produce an aligned subset of the corpus suitable for further research. Thus for higher level section alignment, they used lexical anchors and simply rejected sections that did not align adequately. Using this method on the Canadian Hansard transcripts, they found that sometimes sections appeared in different places in the two languages, and this 'bad' text could simply be ignored. Other differences in the model used need not overly concern us, but we note that they used the EM algorithm to automatically set the various parameters of the model (see section 13.3). They report very good results, at least on 1:1 alignments, but note that sometimes small passages were misaligned because the algorithm ignores the identity of words (just looking at sentence lengths).

Wu (1994)

Wu (1994) begins by applying the method of Gale and Church (1993) to a corpus of parallel English and Cantonese text from the Hong Kong Hansard. He reports that some of the statistical assumptions underlying Gale and Church's model are not as clearly met when dealing with these unrelated languages, but nevertheless, outside of certain header

passages, Wu reports results not much worse than those reported by Gale and Church. To improve accuracy, Wu explores using lexical cues, which heads this work in the direction of the lexical methods that we cover in section 13.1.4. Incidentally, it is interesting to note that Wu's 500 sentence test suite includes one each of a 3:1, 1:3 and 3:3 alignments – alignments considered too exotic to be generable by most of the methods we discuss, including Wu's.

13.1.3 Offset alignment by signal processing techniques

What ties these methods together is that they do not attempt to align beads of sentences but rather just to align position offsets in the two parallel texts so as to show roughly what offset in one text aligns with what offset in the other.

Church (1993)

Church (1993) argues that while the above length-based methods work well on clean texts, such as the Canadian Hansard, they tend to break down in real world situations when one is dealing with noisy optical character recognition (OCR) output, or files that contain unknown markup conventions. OCR programs can lose paragraph breaks and punctuation characters, and floating material (headers, footnotes, tables, etc.) can confuse the linear order of text to be aligned. In such texts, finding even paragraph and sentence boundaries can be difficult. Electronic texts should avoid most of these problems, but may contain unknown markup conventions that need to be treated as noise. Church's approach is to induce an alignment by using cognates. *Cognates* are words that are similar across languages either due to borrowing or common inheritance from a linguistic ancestor, for instance, French *supérieur* and English *superior*. However, rather than considering cognate words (as in Simard et al. (1992)) or finding lexical correspondences (as in the methods to which we will turn next), the procedure works by finding cognates at the level of character sequences. The method is dependent on there being an ample supply of identical character sequences between the source and target languages, but Church suggests that this happens not only in languages with many cognates but in almost any language using the Roman alphabet, since there are usually many proper names and numbers present. He suggests that the method can even work with non-Roman

COGNATES

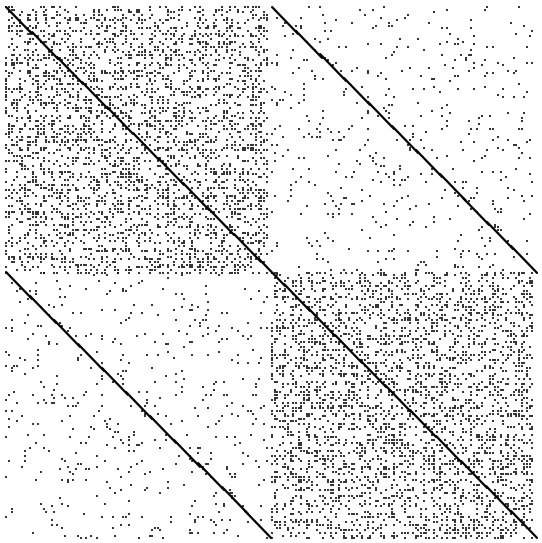

Figure 13.4 A sample dot plot. The source and the translated text are concatenated. Each coordinate (x, y) is marked with a dot iff there is a correspondence between position x and position y. The source text has more random correspondences with itself than with the translated text, which explains the darker shade of the upper left and, by analogy, the darker shade of the lower right. The diagonals are black because there is perfect correspondence of each text with itself (the diagonals in the upper left and the lower right), and because of the correspondences between the source text and its translation (diagonals in lower left and upper right).

writing systems, providing they are liberally sprinkled with names and numbers (or computer keywords!).

DOT-PLOT The method used is to construct a *dot-plot*. The source and translated texts are concatenated and then a square graph is made with this text on both axes. A dot is placed at (x, y) whenever there is a match between positions x and y in this concatenated text. In Church (1993) the unit that is matched is character 4-grams. Various signal processing techniques are then used to compress the resulting plot. Dot plots have a characteristic look, roughly as shown in figure 13.4. There is a straight diagonal line, since each position (x, x) has a dot. There are then two darker rectangles in the upper left and lower right. (Since the source is more similar to itself, and the translation to itself than each to the other.) But the im-

portant information for text alignment is found in the other two, lighter colored quadrants. Either of these matches between the text of the two languages, and hence represents what is sometimes called a *bitext map*. In these quadrants, there are two other, fainter, roughly straight, diagonal lines. These lines result from the propensity of cognates to appear in the two languages, so that often the same character sequences appear in the source and the translation of a sentence. A heuristic search is then used to find the best path along the diagonal, and this provides an alignment in terms of offsets in the two texts. The details of the algorithm need not concern us, but in practice various methods are used so as not to calculate the entire dotplot, and *n*-grams are weighted by inverse frequency so as to give more importance to when rare *n*-grams match (with common *n*-grams simply being ignored). Note that there is no attempt here to align whole sentences as beads, and hence one cannot provide performance figures corresponding to those for most other methods we discuss. Perhaps because of this, the paper offers no quantitative evaluation of performance, although it suggests that error rates are "often very small." Moreover, while this method may often work well in practice, it can never be a fully general solution to the problem of aligning parallel texts, since it will fail completely when no or extremely few identical character sequences appear between the text and the translation. This problem can occur when different character sets are used, as with eastern European or Asian languages (although even in such case there are often numbers and foreign language names that occur on both sides).

BITEXT MAP

Fung and McKeown (1994)

Following earlier work in Fung and Church (1994), Fung and McKeown (1994) seek an algorithm that will work: (i) without having found sentence boundaries (as we noted above, punctuation is often lost in OCR), (ii) in only roughly parallel texts where some sections may have no corresponding section in the translation or vice versa, and (iii) with unrelated language pairs. In particular, they wish to apply this technique to a parallel corpus of English and Cantonese (Chinese). The technique is to infer a small bilingual dictionary that will give points of alignment. For each word, a *signal* is produced, as an *arrival vector* of integer numbers giv-

ARRIVAL VECTOR

ing the number of words between each occurrence of the word at hand.[1] For instance, if a word appears at word offsets (1, 263, 267, 519) then the arrival vector will be (262, 4, 252). These vectors are then compared for English and Cantonese words. If the frequency or position of occurrence of an English and a Cantonese word differ too greatly it is assumed that they cannot match, otherwise a measure of similarity between the signals is calculated using Dynamic Time Warping – a standard dynamic programming algorithm used in speech recognition for aligning signals of potentially different lengths (Rabiner and Juang 1993: sec. 4.7). For all such pairs of an English word and a Cantonese word, a few dozen pairs with very similar signals are retained to give a small bilingual dictionary with which to anchor the text alignment. In a manner similar to Church's dot plots, each occurrence of this pair of words becomes a dot in a graph of the English text versus the Cantonese text, and again one expects to see a stronger signal in a line along the diagonal (producing a figure similar to figure 13.4). This best match between the texts is again found by a dynamic programming algorithm and gives a rough correspondence in offsets between the two texts. This second phase is thus much like the previous method, but this method has the advantages that it is genuinely language independent, and that it is sensitive to lexical content.

13.1.4 Lexical methods of sentence alignment

The previous methods attacked the lack of robustness of the length-based methods in the face of noisy and imperfect input, but they do this by abandoning the goal of aligning sentences, and just aligning text offsets. In this section we review a number of methods which still align beads of sentences like the first methods, but are more robust because they use lexical information to guide the alignment process.

Kay and Röscheisen (1993)

The early proposals of Brown et al. (1991c) and Gale and Church (1993) make little or no use of the actual lexical content of the sentences. However, it seems that lexical information could give a lot of confirmation of alignments, and be vital in certain cases where a string of similar length

1. Since Chinese is not written divided into words, being able to do this depends on an earlier text segmentation phase.

sentences appears in two languages (as often happens in reports when there are things like lists). Kay and Röscheisen (1993) thus use a partial alignment of lexical items to induce the sentence alignment. The use of lexical cues also means the method does not require a prior higher level paragraph alignment.

The method involves a process of convergence where a partial alignment at the word level induces a maximum likelihood alignment at the sentence level, which is used in turn to refine the word level alignment and so on. Word alignment is based on the assumption that two words should correspond if their distributions are the same. The steps are basically as follows:

- Assume the first and last sentences of the texts align. These are the initial anchors.
- Then until most sentences are aligned:

ENVELOPE

1. Form an *envelope* of possible alignments from the cartesian product of the list of sentences in the source language and the target language. Alignments are excluded if they cross anchors or their respective distances from an anchor differ too greatly. The difference is allowed to increase as distance from an anchor increases, giving a pillow shape of possible alignments, as in figure 13.5.
2. Choose pairs of words that tend to co-occur in these potential partial alignments. Choose words whose distributions are similar in the sense that most of the sentences in which one appears are alignable with sentences in which the other appears, and which are sufficiently common that alignments are not likely to be due to chance.
3. Find pairs of source and target sentences which contain many possible lexical correspondences. The most reliable of these pairs are used to induce a set of partial alignments which will be part of the final result. We commit to these alignments, and add them to our list of anchors, and then repeat the steps above.

ANNEALING SCHEDULE

The accuracy of the approach depends on the *annealing schedule*. If you accept many pairs as reliable in each iteration, you need fewer iterations but the results might suffer. Typically, about 5 iterations are needed for satisfactory results. This method does not assume any limitations on the types of possible alignments, and is very robust, in that

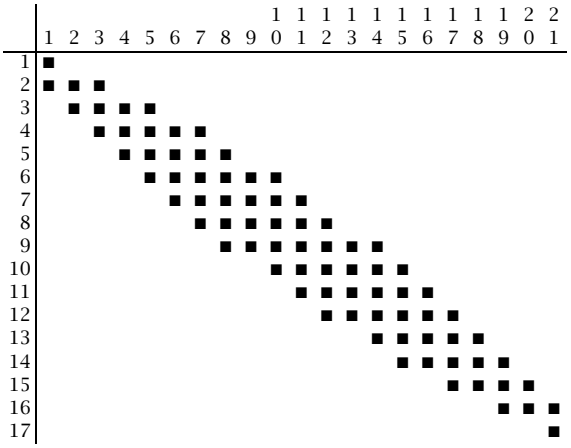

Figure 13.5 The pillow-shaped envelope that is searched. Sentences in the L_1 text are shown on the vertical axis (1–17), sentences in the L_2 text are shown on the horizontal axis (1–21). There is already an anchor between the beginning of both texts, and between sentences $(17, 21)$. A '■' indicates that the two corresponding sentences are in the set of alignments that are considered in the current iteration of the algorithm. Based on (Kay and Röscheisen 1993: figure 3).

'bad' sentences just will not have any match in the final alignment. Results are again good. On Scientific American articles, Kay and Röscheisen (1993) achieved 96% coverage after four passes, and attributed the remainder to 1:0 and 0:1 matches. On 1000 Hansard sentences and using 5 passes, there were 7 errors, 5 of which they attribute not to the main algorithm but to the naive sentence boundary detection algorithm that they employed. On the other hand, the method is computationally intensive. If one begins with a large text with only the endpoints for anchors, there will be a large envelope to search. Moreover, the use of a pillow-shaped envelope to somewhat constrain the search could cause problems if large sections of the text have been moved around or deleted, as then the correct alignments for certain sentences may lie outside the search envelope.

Chen (1993)

Chen (1993) does sentence alignment by constructing a simple word-to-word translation model as he goes along. The best alignment is then

the one that maximizes the likelihood of generating the corpus given the translation model. This best alignment is again found by using dynamic programming. Chen argues that whereas previous length-based methods lacked robustness and previous lexical methods were too slow to be practical for large tasks, his method is robust, fast enough to be practical (thanks to using a simple translation model and thresholding methods to improve the search for the best alignment), and more accurate than previous methods.

The model is essentially like that of Gale and Church (1993), except that a translation model is used to estimate the cost of a certain alignment. So, to align two texts S and T, we divide them into a sequence of sentence beads B_k, each containing zero or more sentences of each language as before, so that the sequence of beads covers the corpus:

$$B_k = (s_{a_k}, \ldots, s_{b_k};\ t_{c_k}, \ldots, t_{d_k})$$

Then, assuming independence between sentence beads, the most probable alignment $A = B_1, \ldots, B_{m_A}$ of the corpus is determined by:

$$\arg\max_A P(S, T, A) = \arg\max_A P(L) \prod_{k=1}^{m_A} P(B_k)$$

The term $P(L)$ is the probability that one generates an alignment of L beads, but Chen effectively ignores this term by suggesting that this distribution is uniform up to some suitably high ℓ greater than the number of sentences in the corpus, and zero thereafter.

The task then is to determine a translation model that gives a more accurate probability estimate, and hence cost for a certain bead than a model based only on the length of the respective sentences. Chen argues that for reasons of simplicity and efficiency one should stick to a fairly simple translation model. The model used ignores issues of word order, and the possibility of a word corresponding to more than one word in the translation. It makes use of word beads, and these are restricted to 1:0, 0:1, and 1:1 word beads. The essence of the model is that if a word is commonly translated by another word, then the probability of the corresponding 1:1 word bead will be high, much higher than the product of the probability of the 1:0 and 0:1 word beads using the same words. We omit the details of the translation model here, since it is a close relative of the model introduced in section 13.3. For the probability of an alignment, the program does not sum over possible word beadings derived from the sentences in the bead, but just takes the best one. Indeed, it does not

WORD BEADS

even necessarily find the best one since it does a greedy search for the best word beading: the program starts with a 1:0 and 0:1 beading of the putative alignment, and greedily replaces a 1:0 and a 0:1 bead with the 1:1 bead that produces the biggest improvement in the probability of the alignment until no further improvement can be gained.

The parameters of Chen's model are estimated by a Viterbi version of the EM algorithm.[2] The model is bootstrapped from a small corpus of 100 sentence pairs that have been manually aligned. It then reestimates parameters using an incremental version of the EM algorithm on an (unannotated) chunk of 20,000 corresponding sentences from each language. The model then finally aligns the corpora using a single pass through the data. The method of finding the best total alignment uses dynamic programming as in Gale and Church (1993). However, thresholding is used for speed reasons (to give a linear search rather than the quadratic performance of dynamic programming): a beam search is used and only partial prefix alignments that are almost as good as the best partial alignment are maintained in the beam. This technique for limiting search is generally very effective, but it causes problems when there are large deletions or insertions in one text (vanilla dynamic programming should be much more robust against such events, but see Simard and Plamondon (1996)). However, Chen suggests it is easy to detect large deletions (the probability of all alignments becomes low, and so the beam becomes very wide), and a special procedure is then invoked to search for a clear alignment after the deletion, and the regular alignment process is then restarted from this point.

This method has been used for large scale alignments: several million sentences each of English and French from both Canadian Hansard and European Economic Community proceedings. Chen has estimated the error rate based on assessment of places where the proposed alignment is different from the results of Brown et al. (1991c). He estimates an error rate of 0.4% over the entire text whereas others have either reported higher error rates or similar error rates over only a subset of the text. Finally Chen suggests that most of the errors are apparently due to the not-terribly-good sentence boundary detection method used, and that further

2. In the standard EM algorithm, for each data item, one sums over all ways of doing something to get an expectation for a parameter. Sometimes, for computational reasons, people adopt the expedient of just using the probabilities of the best way of doing something for each data item instead. This method is referred to as a Viterbi version of the EM algorithm. It is heuristic, but can be reasonably effective.

improvements in the translation model are unlikely to improve the alignments, while tending to make the alignment process much slower. We note, however, that the presented work limits matches to 1:0, 0:1, 1:1, 2:1, and 1:2, and so it will fail to find the more exotic alignments that do sometimes occur. Extending the model to other alignment types appears straightforward, although we note that in practice Gale and Church had less success in finding unusual alignment types. Chen does not present any results broken down according to the type of alignment involved.

Haruno and Yamazaki (1996)

Haruno and Yamazaki (1996) argue that none of the above methods work effectively when trying to align short texts in structurally different languages. Their proposed method is essentially a variant of Kay and Röscheisen (1993), but nevertheless, the paper contains several interesting observations.[3] Firstly they suggest that for structurally very different languages like Japanese and English, including function words in lexical matching actually impedes alignment, and so the authors leave all function words out and do lexical matching on content words only. This is achieved by using part of speech taggers to classify words in the two languages. Secondly, if trying to align short texts, there are not enough repeated words for reliable alignment using the techniques Kay and Röscheisen (1993) describe, and so they use an online dictionary to find matching word pairs. Both these techniques mark a move from the knowledge-poor approach that characterized early Statistical NLP work to a knowledge-rich approach. For practical purposes, since knowledge sources like taggers and online dictionaries are widely available, it seems silly to avoid their use purely on ideological grounds. On the other hand, when dealing with more technical texts, Haruno and Yamazaki point out that finding word correspondences in the text is still important – using a dictionary is not a substitute for this. Thus, using a combination of methods they are able to achieve quite good results on even short texts between very different languages.

3. On the other hand some of the details of their method are questionable: use of mutual information to evaluate word matching (see the discussion in section 5.4 – adding use of a t score to filter the unreliability of mutual information when counts are low is only a partial solution) and the use of an ad hoc scoring function to combine knowledge from the dictionary with corpus statistics.

13.1.5 Summary

The upshot seems to be that if you have clean texts from a controlled translation environment, sentence alignment does not seem that difficult a problem, and there are now many methods that perform well. On the other hand, real world problems and less literal translations, or languages with few cognates and different writing systems can pose considerable problems. Methods that model relationships between lexical items in one way or another are much more general and robust in circumstances of this sort. Both signal processing techniques and whole sentence alignment techniques are crude approximations to the fine-grained structure of a match between a sentence and its translation (compare, again, the elaborate microstructure of the match shown in figure 13.2), but they have somewhat different natures. The choice of which to use should be determined by the languages of interest, the required accuracy, and the intended application of the text alignment.

13.1.6 Exercises

Exercise 13.1 [⋆]

For two languages you know, find an example where the basic assumption of the length-based approach breaks down, that is a short and a long sentence are translations of each other. It is easier to find examples if length is defined as number of words.

Exercise 13.2 [⋆]

Gale and Church (1993) argue that measuring length in number of characters is preferable because the variance in number of words is greater. Do you agree that word-based length is more variable? Why?

Exercise 13.3 [⋆]

The dotplot figure is actually incorrect: it is not symmetric with respect to the main diagonal. (Verify this!) It should be. Why?

13.2 Word Alignment

BILINGUAL DICTIONARIES
TERMINOLOGY DATABASES

A common use of aligned texts is the derivation of bilingual dictionaries and terminology databases. This is usually done in two steps. First the text alignment is extended to a word alignment (unless we are dealing with an approach in which word and text alignment are induced simultaneously). Then some criterion such as frequency is used to select aligned

pairs for which there is enough evidence to include them in the bilingual dictionary. For example, if there is just one instance of the word alignment "*adeptes* – *products*" (an alignment that might be derived from figure 13.2), then we will probably not include it in a dictionary (which is the right decision here since *adeptes* means 'users' in the context, not 'products').

ASSOCIATION

One approach to word alignment was briefly discussed in section 5.3.3: word alignment based on measures of association. Association measures such as the χ^2 measure used by Church and Gale (1991b) are an efficient way of computing word alignments from a bitext. In many cases, they are sufficient, especially if a high confidence threshold is used. However, association measures can be misled in situations where a word in L_1 frequently occurs with more than one word in L_2. This was the example of *house* being incorrectly translated as *communes* instead of *chambre* because, in the Hansard, *House* most often occurs with both French words in the phrase *Chambre de Communes*.

Pairs like *chambre*↔*house* can be identified if we take into account a source of information that is ignored by pure association measures: the fact that, on average, a given word is the translation of only one other word in the second language. Of course, this is true for only part of the words in an aligned text, but assuming one-to-one correspondence has been shown to give highly accurate results (Melamed 1997b). Most algorithms that incorporate this type of information are implementations of the EM algorithm or involve a similar back-and-forth between a hypothesized dictionary of word correspondences and an alignment of word tokens in the aligned corpus. Examples include Chen (1993) as described in the previous section, Brown et al. (1990) as described in the next section, Dagan et al. (1993), Kupiec (1993a), and Vogel et al. (1996). Most of these approaches involve several iterations of recomputing word correspondences from aligned tokens and then recomputing the alignment of tokens based on the improved word correspondences. Other authors address the additional complexity of deriving correspondences between

PHRASES

phrases since in many cases the desired output is a database of terminological expressions, many of which can be quite complex (Wu 1995; Gaussier 1998; Hull 1998). The need for several iterations makes all of these algorithms somewhat less efficient than pure association methods.

As a final remark, we note that future work is likely to make significant use of the prior knowledge present in existing bilingual dictionaries rather than attempting to derive everything from the aligned text. See

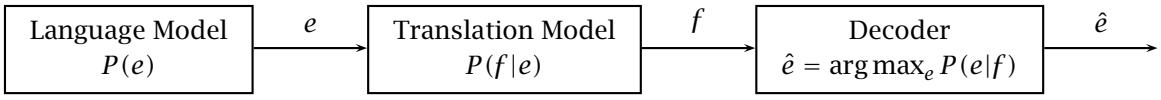

Figure 13.6 The noisy channel model in machine translation. The Language Model generates an English sentence e. The Translation Model transmits e as the French sentence f. The decoder finds the English sentence $\hat{e}$ which is most likely to have given rise to f.

Klavans and Tzoukermann (1995) for one example of such an approach.

13.3 Statistical Machine Translation

NOISY CHANNEL MODEL

In section 2.2.4, we introduced the *noisy channel model*. One of its applications in NLP is machine translation as shown in figure 13.6. In order to translate from French to English, we set up a noisy channel that receives as input an English sentence e, transforms it into a French sentence f, and sends the French sentence f to a decoder. The decoder then determines the English sentence $\hat{e}$ that f is most likely to have arisen from (and which is not necessarily identical to e).

We thus have to build three components for translation from French to English: a language model, a translation model, and a decoder. We also have to estimate the parameters of the model, the *translation probabilities*.

Language model. The language model gives us the probability $P(e)$ of the English sentence. We already know how to build language models based on n-grams (chapter 6) or probabilistic grammars (chapter 11 and chapter 12), so we just assume here that we have an appropriate language model.

Translation model. Here is a simple translation model based on word alignment:

$$P(f|e) = \frac{1}{Z} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} P(f_j|e_{a_j}) \tag{13.5}$$

We use the notation of Brown et al. (1993): e is the English sentence; l is the length of e in words; f is the French sentence; m is the length of f;

TRANSLATION PROBABILITY

f_j is word j in f; a_j is the position in e that f_j is aligned with; e_{a_j} is the word in e that f_j is aligned with; $P(w_f|w_e)$ is the *translation probability*, the probability that we will see w_f in the French sentence given that we see w_e in the English sentence; and Z is a normalization constant.

EMPTY CEPT

The basic idea of this formula is fairly straightforward. The m sums $\sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l}$ sum over all possible alignments of French words to English words. The meaning of $a_j = 0$ for an a_j is that word j in the French sentence is aligned with the *empty cept*, that is, it has no (overt) translation. Note that an English word can be aligned with multiple French words, but that each French word is aligned with at most one English word.

For a particular alignment, we multiply the m translation probabilities, assuming independence of the individual translations (see below for how to estimate the translation probabilities). So for example, if we want to compute:

$$P(\textit{Jean aime Marie}|\textit{John loves Mary})$$

for the alignment (*Jean*, *John*), (*aime*, *loves*), and (*Marie*, *Mary*), then we multiply the three corresponding translation probabilities.

$$P(\textit{Jean}|\textit{John}) \times P(\textit{aime}|\textit{loves}\ ) \times P(\textit{Marie}|\textit{Mary})$$

To summarize, we compute $P(f|e)$ by summing the probabilities of all alignments. For each alignment, we make two (rather drastic) simplifying assumptions: Each French word is generated by exactly one English word (or the empty cept); and the generation of each French word is independent of the generation of all other French words in the sentence.[4]

Decoder. We saw examples of decoders in section 2.2.4 and this one does the same kind of maximization, based on the observation that we can omit $P(f)$ from the maximization since f is fixed:

$$\hat{e} = \arg\max_e P(e|f) = \arg\max_e \frac{P(e)P(f|e)}{P(f)} = \arg\max_e P(e)P(f|e) \tag{13.6}$$

The problem is that the search space is infinite, so we need a heuristic search algorithm. One possibility is to use stack search (see section 12.1.10). The basic idea is that we build an English sentence incrementally. We keep a stack of partial translation hypotheses. At each

4. Going in the other direction, note that one English word can correspond to multiple French words.

point, we extend these hypotheses with a small number of words and alignments and then prune the stack to its previous size by discarding the least likely extended hypotheses. This algorithm is not guaranteed to find the best translation, but can be implemented efficiently.

Translation probabilities. The translation probabilities are estimated using the EM algorithm (see section 14.2.2 for a general introduction to EM). We assume that we have a corpus of aligned sentences.

As we discussed in the previous section on word alignment, one way to guess at which words correspond to each other is to compute an association measure like χ^2. But that will generate many spurious correspondences because a source word is not penalized for being associated with more than one target word (recall the example *chambre↔house*, *chambre↔chamber*).

CREDIT ASSIGNMENT

The basic idea of the EM algorithm is that it solves the *credit assignment* problem. If a word in the source is strongly aligned with a word in the target, then it is not available anymore to be aligned with other words in the target. This avoids cases of double and triple alignment on the one hand, and an excessive number of unaligned words on the other hand.

We start with a random initialization of the translation probabilities $P(w_f|w_e)$. In the E step, we compute the expected number of times we will find w_f in the French sentence given that we have w_e in the English sentence.

$$z_{w_f,w_e} = \sum_{(e,f) \text{ s.t. } w_e \in e, w_f \in f} P(w_f|w_e)$$

where the summation ranges over all pairs of aligned sentences such that the English sentence contains w_e and the French sentence contains w_f. (We have simplified slightly here since we ignore cases where words occur more than once in a sentence.)

The M step reestimates the translation probabilities from these expectations:

$$P(w_f|w_e) = \frac{z_{w_f,w_e}}{\sum_v z_{w_f,v}}$$

where the summation ranges over all English words v.

What we have described is a very simple version of the algorithms described by Brown et al. (1990) and Brown et al. (1993) (see also Kupiec (1993a) for a clear statement of EM for alignment). The main part we

have simplified is that, in these models, implausible alignments are penalized. For example, if an English word at the beginning of the English sentence is aligned with a French word at the end of the French sentence, then this *distortion* in the positions of the two aligned words will decrease the probability of the alignment.

DISTORTION

FERTILITY

Similarly, a notion of *fertility* is introduced for each English word which tells us how many French words it usually generates. In the unconstrained model, we do not distinguish the case where each French word is generated by a different English word, or at least approximately so (which somehow seems the normal case) from the case where all French words are generated by a single English word. The notion of fertility allows us to capture the tendency of word alignments to be one-to-one and one-to-two in most cases (and one-to-zero is another possibility in this model). For example, the most likely fertility of *farmers* in the corpus that the models were tested on is 2 because it is most often translated as two words: *les agriculteurs*. For most English words, the most likely fertility is 1 since they tend to be translated by a single French word.

An evaluation of the model on the aligned Hansard corpus found that only about 48% of French sentences were decoded (or translated) correctly. The errors were either incorrect decodings as in (13.7) or ungrammatical decodings as in (13.8) (Brown et al. 1990: 84).

(13.7) a. **Source sentence.** Permettez que je donne un example à la chambre.

b. **Correct translation.** Let me give the House one example.

c. **Incorrect decoding.** Let me give an example in the House.

(13.8) a. **Source sentence.** Vous avez besoin de toute l'aide disponible.

b. **Correct translation.** You need all the help you can get.

c. **Ungrammatical decoding.** You need of the whole benefits available.

A detailed analysis in (Brown et al. 1990) and (Brown et al. 1993) reveals several problems with the model.

- **Fertility is asymmetric.** Often a single French word corresponds to several English words. For example, *to go* is translated as *aller*. There is no way to capture this generalization in the formalization proposed. The model can get individual sentences with *to go* right by translating

to as the empty set and *go* as *aller*, but this is done in an error-prone way on a case by case basis instead of noting the general correspondence of the two expressions.

Note that there is an asymmetry here since we can formalize the fact that a single English word corresponds to several French words. This is the example of *farmers* which has fertility 2 and produces two words *les* and *agriculteurs*.

- **Independence assumptions.** As so often in Statistical NLP, many independence assumptions are made in developing the probabilistic model that don't strictly hold. As a result, the model gives an unfair advantage to short sentences because, simply put, fewer probabilities are multiplied and therefore the resulting likelihood is a larger number. One can fix this by multiplying the final likelihood with a constant c^l that increases with the length l of the sentence, but a more principled solution would be to develop a more sophisticated model in which inappropriate independence assumptions need not be made. See Brown et al. (1993: 293), and also the discussion in section 12.1.11.

- **Sensitivity to training data.** Small changes in the model and the training data (e.g., taking the training data from different parts of the Hansard) can cause large changes in the estimates of the parameters. For example, the 1990 model has a translation probability $P(le|the)$ of 0.610, the 1993 model has 0.497 instead (Brown et al. 1993: 286). It does not necessarily follow that such discrepancies would impact translation performance negatively, but they certainly raise questions about how close the training text and the text of application need to be in order to get acceptable results. See section 10.3.2 for a discussion of the effect of divergence between training and application corpora in the case of part-of-speech tagging.

- **Efficiency.** Sentences of more than 30 words had to be eliminated from the training set presumably because decoding them took too long (Brown et al. 1993: 282).

On the surface, these are problems of the model, but they are all related to the lack of linguistic knowledge in the model. For example, syntactic analysis would make it possible to relate subparts of the sentence to each other instead of simulating such relations inadequately using the notion of fertility. And a stronger model would make fewer independence

assumptions, make better use of the training data (since a higher bias reduces variance in parameter estimates) and reduce the search space with potential benefits for efficiency in decoding.

Other problems found by Brown et al. (1990) and Brown et al. (1993) show directly that the lack of linguistic knowledge encoded in the system causes many translation failures.

- **No notion of phrases.** The model relates only individual words. As the examples of words with high fertility show, one should really model relationships between phrases, for example, the relationship between *to go* and *aller* and between *farmers* and *les agriculteurs*.

- **Non-local dependencies.** Non-local dependencies are hard to capture with 'local' models like n-gram models (see page 98 in chapter 3). So even if the translation model generates the right set of words, the language model will not assemble them correctly (or will give the reassembled sentence a low probability) if a long-distance dependency occurs. In later work that builds on the two models we discuss here, sentences are preprocessed to reduce the number of long-distance dependencies in order to address this problem (Brown et al. 1992a). For example, *is she a mathematician* would be transformed to *she is a mathematician* in a preprocessing step.

- **Morphology.** Morphologically related words are treated as separate symbols. For example, the fact that each of the 39 forms of the French verb *diriger* can be translated as *to conduct* and *to direct* in appropriate contexts has to be learned separately for each form.

- **Sparse data problems.** Since parameters are solely estimated from the training corpus without any help from other sources of information about words, estimates for rare words are unreliable. Sentences with rare words were excluded from the evaluation in (Brown et al. 1990) because of the difficulty of deriving a good characterization of infrequent words automatically.

In summary, the main problem with the noisy channel model that we have described here is that it incorporates very little domain knowledge about natural language. This is an argument that is made in both (Brown et al. 1990) and (Brown et al. 1993). All subsequent work on statistical machine translation (starting with Brown et al. (1992a)) has therefore

focussed on building models that formalize the linguistic regularities inherent in language.

Non-linguistic models are fairly successful for word alignment as shown by Brown et al. (1993) among others. The research results we have discussed in this section suggest that they fail for machine translation.

Exercise 13.4 [⋆⋆]

The model's task is to find an English sentence given a French input sentence. Why don't we just estimate $P(e|f)$ and do without a language model? What would happen to ungrammatical French sentences if we relied on $P(e|f)$? What happens with ungrammatical French sentences in the model described above that relies on $P(f|e)$? These questions are answered by (Brown et al. 1993: 265).

Exercise 13.5 [⋆]

Translation and fertility probabilities tell us which words to generate, but not where to put them. Why do the generated words end up in the right places in the decoded sentence, at least most of the time?

Exercise 13.6 [⋆⋆]

VITERBI TRANSLATION

The *Viterbi translation* is defined as the translation resulting from the maximum likelihood alignment. In other words, we don't sum over all possible alignments as in the translation model in equation (13.5). Would you expect there to be significant differences between the Viterbi translation and the best translation according to equation (13.5)?

Exercise 13.7 [⋆⋆]

Construct a small training example for EM and compute at least two iterations.

Exercise 13.8 [⋆⋆]

For the purposes of machine translation, n-gram models are reasonable language models for short sentences. However, with increasing sentence length it becomes more likely that there are several (semantically distinct) ways of ordering the words into a grammatical sentence. Find a set of (a) 4 English words, (b) 10 English words that can be turned into two semantically distinct and grammatical sequences.

13.4 Further Reading

For more background on statistical methods in MT, we recommend the overview article by Knight (1997). Readers interested in efficient decoding algorithms (in practice one of the hardest problems in statistical MT) should consult Wu (1996), Wang and Waibel (1997), and Nießen et al. (1998). Alshawi et al. (1997), Wang and Waibel (1998), and Wu and Wong

(1998) attempt to replace the statistical word-for-word approach with a statistical transfer approach (in the terminology of figure 13.1). An algorithm for statistical generation is proposed by Knight and Hatzivassiloglou (1995).

EXAMPLE-BASED

An 'empirical' approach to MT that is different from the noisy channel model we have covered here is *example-based* translation. In example-based translation, one translates a sentence by using the closest match in an aligned corpus as a template. If there is an exact match in the aligned corpus, one can just retrieve the previous translation and be done. Otherwise, the previous translation needs to be modified appropriately. See Nagao (1984) and Sato (1992) for descriptions of example-based MT systems.

TRANSLITERATION

One purpose of word correspondences is to use them in translating unknown words. However, even with automatic acquisition from aligned corpora, there will still be unknown words in any new text, in particular names. This is a particular problem when translating between languages with different writing systems since one cannot use the unknown string verbatim in the translation of, say, Japanese to English. Knight and Graehl (1997) show how many proper names can be handled by a *transliteration* system that infers the written form of the name in the target language directly from the written form in the source language. Since the roman alphabet is transliterated fairly systematically into character sets like Cyrillic, the original Roman form can often be completely recovered.

Finding word correspondences can be seen as a special case of the more general problem of knowledge acquisition for machine translation. See Knight et al. (1995) for a more high-level view of acquisition in the MT context that goes beyond the specific problems we have discussed here.

Using parallel texts as a knowledge source for word sense disambiguation is described in Brown et al. (1991b) and Gale et al. (1992d) (see also section 7.2.2). The example of the use of text alignment as an aid for translators revising product literature is taken from Shemtov (1993). The alignment example at the beginning of the chapter is drawn from an example text from the UBS data considered by Gale and Church (1993), although they do not discuss the word level alignment. Note that the text in both languages is actually a translation from a German original. A search interface to examples of aligned French and English Canadian Hansard sentences is available on the web; see the website.

BEAD
BITEXT
BITEXT MAP

The term *bead* was introduced by Brown et al. (1991c). The notion of a *bitext* is from (Harris 1988), and the term *bitext map* comes from

(Melamed 1997a). Further work on signal-processing-based approaches to parallel text alignment appears in (Melamed 1997a) and (Chang and Chen 1997). A recent evaluation of a number of alignment systems is available on the web (see website). Particularly interesting is the very divergent performance of the systems on different parallel corpora presenting different degrees of difficulty in alignment.