The Mathematics of Machine Translation: Parameter Estimation

Peter F. Brown, Stephen A. Della Pietra
Vincent J. Della Pietra, Robert L. Mercer

The availability of large, bilingual corpora has stimulated recent interest in algorithms for manipulating them. A number of authors have discussed algorithms for extracting from such corpora pairs of sentences that are translations of one another. In the course of our work on machine translation, we have developed a series of five statistical models of the translation process. Here, we describe these models and show that it is possible to estimate their parameters automatically from a large set of pairs of sentences. We show, further, that it is possible to align the words within pairs of sentences algorithmically. We have a great deal of data in French and English from the proceedings of the Canadian Parliament. For this reason we have restricted our work to these two languages, but we feel that because our algorithms have minimal linguistic content they would work well on other pairs of languages. We also feel, again because of the minimal linguistic content of our algorithms, that it is reasonable to argue that word-by-word alignments are inherent in any sufficiently large bilingual corpora.

1.1 Introduction

The growing availability of bilingual, machine-readable texts has stimulated interest in methods for extracting linguistically valuable information from such texts. For example, a number of recent papers deal with the problem of automatically obtaining pairs of aligned sentences from parallel corpora [95, 28, 52, 65]. Brown *et al.* [27] assert, and Brown, Lai, and Mercer [28] and Gale and Church [52] both show, that it is possible to obtain such aligned pairs of sentences without inspecting the words that the sentences contain. Brown, Lai, and Mercer base their algorithm on the number of words that the sentences contain while Gale and Church base a similar algorithm on the number of characters that the sentences contain. The lesson to be learned from these two efforts is that simple, statistical methods can be surprisingly successful

This paper first appeared in *Computational Linguistics*, vol. 19, number 2. June, 1993. pp.263–311

in achieving linguistically interesting goals. Here, we address a natural extension of that work: matching up the words within pairs of aligned sentences.

In recent papers, Brown *et al.* [26, 27] propose a statistical approach to machine translation from French to English. In the latter of these papers, they sketch an algorithm for estimating the probability that an English word will be translated into any particular French word and show that such probabilities, once estimated, can be used together with a statistical model of the translation process to align the words in an English sentence with the words in its French translation (see their Figure 3). Pairs of sentences with words aligned in this way offer a valuable resource for work in bilingual lexicography and machine translation.

The next Section is a synopsis of our statistical approach to machine translation. Following this synopsis, we develop some terminology and notation for describing the word-by-word alignment of pairs of sentences. In Section 1.4 we describe our series of models of the translation process and give an informal discussion of the algorithms by which we estimate their parameters from data. We have written Section 1.4 with two aims in mind. First, we wished to provide the interested reader with sufficient detail that he would be able to reproduce our results if he had access to a modern workstation and to the necessary bilingual data; and second, we wished to hold the mathematics at an elementary level so that a reader who is proficient at college calculus, specifically Lagrange multipliers and partial derivatives, would be able to work his way through it. A few more difficult parts of the discussion have been postponed to the Appendix.

In Section 1.5, we present results obtained by estimating the parameters for these models from a large collection of aligned pairs of sentences from the Canadian Hansard data [28]. For a number of English words, we show translation probabilities that give convincing evidence of the power of automatic statistical methods to extract linguistically interesting correlations from large corpora. We also show automatically derived word-by-word alignments for several sentences.

In Section 1.6, we discuss some shortcomings of our models and propose modifications to address some of them. In the last section, we discuss the significance of our work and the possibility of extending it to other pairs of languages.

Finally, we include two Appendices: one to summarize notation, and one to collect the formulae for the various models that we describe and to fill an occasional gap in our development.

1.2 Statistical Translation

In 1949, Warren Weaver suggested applying the statistical and cryptanalytic techniques then emerging from the nascent field of communication theory to the problem of using computers to translate text from one natural language to another [96]. Efforts in this direction were soon abandoned for various philosophical and theoretical reasons, but at a time when the most advanced computers were of a piece with today's digital watch, any such approach was surely doomed to computational starvation. The extraordinary progress in computer technology over the last four decades makes it prudent to look once again at this type of approach. Today, the fruitful application of statistical methods to the study of machine translation is within the computational grasp of anyone with a well-equipped workstation.

A string of English words, $\mathbf{e}$, can be translated into a string of French words in many different ways. Often, knowing the broader context in which $\mathbf{e}$ occurs may serve to winnow the field of acceptable French translations, but even so, many acceptable translations will remain; the choice among them is largely a matter of taste. In statistical translation, we take the view that every French string, $\mathbf{f}$, is a possible translation of $\mathbf{e}$. We assign to every pair of strings $(\mathbf{e}, \mathbf{f})$ a number $\Pr(\mathbf{f}|\mathbf{e})$, which we interpret as the probability that a translator, when presented with $\mathbf{e}$ will produce $\mathbf{f}$ as his translation. We further take the view that when a native speaker of French produces a string of French words, he has actually conceived of a string of English words, which he translated mentally. Given a French string $\mathbf{f}$, the job of our translation system is to find the string $\mathbf{e}$ that the native speaker had in mind when he produced $\mathbf{f}$. We minimize our chance of error by choosing that English string $\hat{\mathbf{e}}$ for which $\Pr(\mathbf{e}|\mathbf{f})$ is greatest.

Using Bayes' theorem, we can write

$$\Pr(\mathbf{e}|\mathbf{f}) = \frac{\Pr(\mathbf{e})\Pr(\mathbf{f}|\mathbf{e})}{\Pr(\mathbf{f})}. \tag{1.1}$$

Since the denominator here is independent of $\mathbf{e}$, finding $\hat{\mathbf{e}}$ is the same as finding $\mathbf{e}$ so as to make the product $\Pr(\mathbf{e})\Pr(\mathbf{f}|\mathbf{e})$ as large as possible. We arrive, then, at the Fundamental Equation of Statistical Machine Translation:

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{e})\Pr(\mathbf{f}|\mathbf{e}). \tag{1.2}$$

As a representation of the process by which a human being translates a passage from French to English, this equation is fanciful at best. One can hardly imagine someone rifling mentally through the list of all English passages computing the product of the *a priori* probability of the passage, $\Pr(\mathbf{e})$, and the conditional probability of the French passage given the English passage, $\Pr(\mathbf{f}|\mathbf{e})$. Instead, there is an overwhelming intuitive appeal to the idea that a translator proceeds by first understanding the French, and then expressing in English the meaning that he has thus grasped. Many people have been guided by this intuitive picture when building machine translation systems.

From a purely formal point of view, on the other hand, Equation (1.2) is completely adequate. The conditional distribution $\Pr(\mathbf{f}|\mathbf{e})$ is nothing less than an enormous table that associates a real number between zero and one with every possible pairing of a French passage and an English passage. With the proper choice for this distribution, translations of arbitrarily high quality can be achieved. Of course, to construct $\Pr(\mathbf{f}|\mathbf{e})$ by examining individual pairs of French and English passages one by one is out of the question. Even if we restrict our attention to passages no longer than a typical novel, there are just too many such pairs. But this is only a problem in practice not in principal. The essential question for statistical translation, then, is not a philosophical one, but an empirical one: Can one construct approximations to the distributions $\Pr(\mathbf{e})$ and $\Pr(\mathbf{f}|\mathbf{e})$ that are at once good enough to acheive an acceptable quality of translation and yet tractable enough to allow an acceptable speed of translation?

Equation (1.2) summarizes the three computational challenges presented by the practice of statistical translation: estimating the *language model probability,* $\Pr(\mathbf{e})$; estimating the *translation model probability,* $\Pr(\mathbf{f}|\mathbf{e})$; and devising an effective and efficient suboptimal search for the English string that maximizes their product. We call these the language modelling problem, the translation modelling problem, and the search problem.

The language modelling problem for machine translation is essentially the same as that for speech recognition and has been dealt with elsewhere in that context (see, for example, the recent paper by Maltese and Mancini [74] and references therein). We hope to deal with the search problem in a later paper. In this paper, we focus on the translation modelling problem. Before we turn to this problem, however, we should address an issue that may be a concern to some readers: Why do we estimate $\Pr(\mathbf{e})$ and $\Pr(\mathbf{f}|\mathbf{e})$ rather than estimate $\Pr(\mathbf{e}|\mathbf{f})$ directly? We are really interested in this latter probability. Wouldn't we reduce our problems from three to two by this direct approach? If we can estimate $\Pr(\mathbf{f}|\mathbf{e})$ adequately, why can't we just turn the whole process around to estimate $\Pr(\mathbf{e}|\mathbf{f})$?

To understand this, imagine that we divide French and English strings into those that are well-formed and those that are ill-formed. This is not a precise notion. We have in mind that strings like *Il va à la bibliothèque,* or *I live in a house,* or even *Colorless green ideas sleep furiously* are well-formed, but that strings like *à la va Il bibliothèque* or *a I in live house* are not. When we translate a French string into English, we can think of ourselves as springing from a well-formed French string into the sea of well-formed English strings with the hope of landing on a good one. It is important, therefore, that our model for $\Pr(\mathbf{e}|\mathbf{f})$ concentrate its probability as much as possible on well-formed English strings. But it is not important that our model for $\Pr(\mathbf{f}|\mathbf{e})$ concentrate its probability on well-formed French strings. If we were to reduce the probability of all well-formed French strings by the same factor, spreading the probability thus liberated over ill-formed French strings, there would be no effect on our translations: the argument that maximizes some function $f(x)$ also maximizes $cf(x)$ for any positive constant c. As we shall see below, our translation models are prodigal, spraying probability all over the place, most of it on ill-formed French strings. In fact, as we discuss in Section 1.4.5, two of our models waste much of their probability on things that are not strings at all, having, for example, several different second words but no first word. If we were to turn one of these models around to model $\Pr(\mathbf{e}|\mathbf{f})$ directly, the result would be a model with so little probability concentrated on well-formed English strings as to confound any scheme to discover one.

The two factors in Equation (1.2) cooperate. The translation model probability is large for English strings, whether well- or ill-formed, that have the necessary words in them in roughly the right places to explain the French. The language model probability is large for well-formed English strings regardless of their connection to the French. Together, they produce a large probability for well-formed English strings that account well for the French. We cannot achieve this simply by reversing our translation models.

1.3 Alignments

We say that a pair of strings that are translations of one another forms a *translation,* and we show this by enclosing the strings in parentheses and separating them by a vertical bar. Thus, we write the translation *(Qu'aurions-nous pu faire? | What could we have done?)* to show that *What could we have done?* is a translation of *Qu'aurions-nous pu faire?* When the strings end in sentences, we usually omit the final stop unless it is a question mark or an exclamation point.

Brown *et al.* [27] introduce the idea of an alignment between a pair of strings as an object indicating for each word in the French string that word in the English string from which it arose. Alignments are shown graphically, as in Figure 1.1, by writing the English string above the French string and drawing lines, which we call *connections,* from some of the English words to some of the French words. The alignment in Figure 1.1 has seven connections, *(the, Le), (program, programme),* and so on. Following the notation of Brown *et al.*, we write this alignment as *(Le programme a été mis en application | And the(1) program(2) has(3) been(4) implemented(5,6,7)).* The list of numbers following an English word shows the positions in the French string of the words to which it is connected. Because *And* is not connected to any French words here, there is no list of numbers after it. We consider every alignment to be correct with some probability, and so we find *(Le programme a été mis en application | And(1,2,3,4,5,6,7) the program has been implemented)* perfectly acceptable. Of course, we expect it to be much less probable than the alignment shown in Figure 1.1.

In Figure 1.1 each French word is connected to exactly one English word, but more general alignments are possible and may be appropriate for some translations. For example, we may have a French word connected to several English words as in Figure 1.2, which we write as *(Le reste appartenait aux autochtones | The(1) balance(2) was(3) the(3) territory(3) of(4) the(4) aboriginal(5) people(5)).* More generally still, we may have several French words connected to several English words as in Figure 1.3, which we write as *(Les pauvres sont démunis | The(1)*

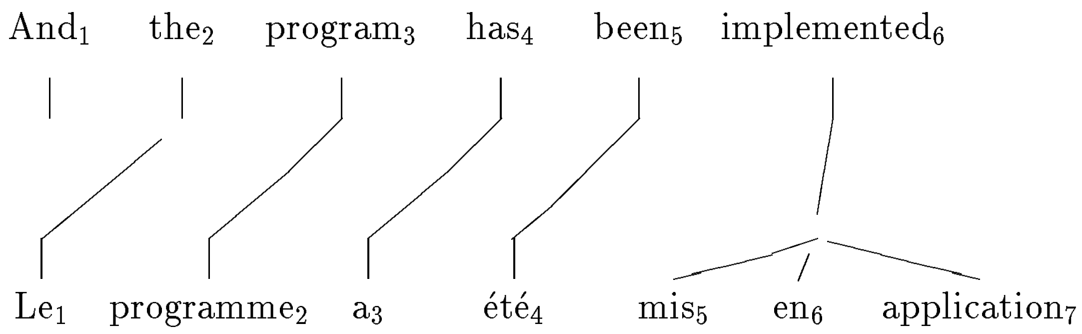

Figure 1.1: An alignment with independent English words.

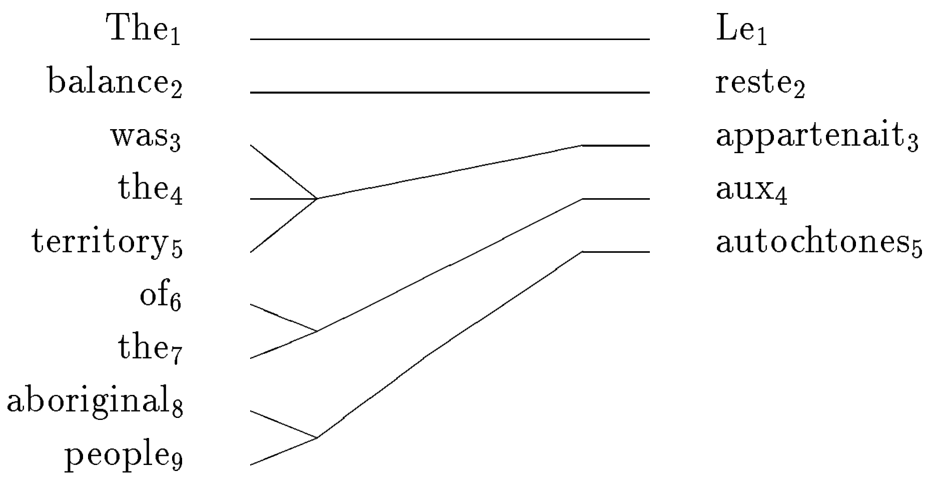

Figure 1.2: An alignment with independent French words.

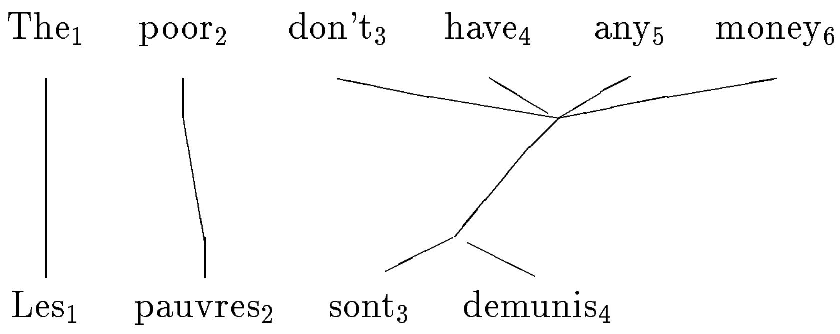

Figure 1.3: A general alignment.

poor(2) don't(3,4) have(3,4) any(3,4) money(3,4)). Here, the four English words *don't have any money* work together to generate the two French words *sont démunis.*

In a figurative sense, an English passage is a web of concepts woven together according to the rules of English grammar. When we look at a passage, we cannot see the concepts directly but only the words that they leave behind. To show that these words are related to a concept but are not quite the whole story, we say that they form a *cept.* Some of the words in a passage may participate in more than one cept, while others may participate in none, serving only as a sort of syntactic glue to bind the whole together. When a passage is translated into French, each of its cepts contributes some French words to the translation. We formalize this use of the term cept and relate it to the idea of an alignment as follows.

We call the set of English words connected to a French word in a particular alignment the cept that generates the French word. Thus, an alignment resolves an English string into a set of possibly overlapping cepts that we call the *ceptual scheme* of the English string with respect to the alignment. The alignment in Figure 1.3 contains the three cepts *The, poor,* and *don't have any money.* When one or more of the French words is connected to no English words, we say that the ceptual scheme includes the empty cept and that each of these words has been generated by this empty cept.

Formally, a cept is a subset of the positions in the English string together with the words occupying those positions. When we write the words that make up a cept, we sometimes affix a subscript to each one showing its position. The alignment in Figure 1.2 includes the cepts the_4 and $of_6\ the_7$, but not the cepts $of_6\ the_4$ or the_7. In *(J'applaudis à la décision* | *I(1) applaud(2) the(4) decision(5)),* *à* is generated by the empty cept. Although the empty cept has no position, we place it by convention in position zero, and write it as e_0. Thus, we may also write the previous alignment as *(J'applaudis à la décision* | e_0*(3) I(1) applaud(2) the(4) decision(5)).*

We denote the set of alignments of $(\mathbf{f}|\mathbf{e})$ by $\mathcal{A}(\mathbf{e},\mathbf{f})$. If $\mathbf{e}$ has length l and $\mathbf{f}$ has length m, there are lm different connections that can be drawn between them because each of the m French words can be connected to any of the l English words. Since an alignment is determined by the connections that it contains, and since a subset of the possible connections can be chosen in 2^{lm} ways, there are 2^{lm} alignments in $\mathcal{A}(\mathbf{e},\mathbf{f})$.

1.4 Translation Models

In this Section, we develop a series of five translation models together with the algorithms necessary to estimate their parameters. Each model gives a prescription for computing the conditional probability $\Pr(\mathbf{f}|\mathbf{e})$, which we call the likelihood of the translation $(\mathbf{f},\mathbf{e})$. This likelihood is a function of a large number of free parameters that we must estimate in a process that we call *training.* The likelihood of a set of translations is the product of the likelihoods of its members. In broad outline, our plan is to guess values for these parameters and then to apply the EM algorithm [15, 46] iteratively so as to approach a local maximum of the likelihood of a particular set of translations that we call the *training data.* When the likelihood of the training data has more than one local maximum, the one that we approach will depend on our initial guess.

In Models 1 and 2, we first choose a length for the French string, assuming all reasonable lengths to be equally likely. Then, for each position in the French string, we decide how to connect it to the English string and what French word to place there. In Model 1 we assume all connections for each French position to be equally likely. Therefore, the order of the words in $\mathbf{e}$ and $\mathbf{f}$ does not affect $\Pr(\mathbf{f}|\mathbf{e})$. In Model 2 we make the more realistic assumption that the probability of a connection depends in the positions which it connects and on the lengths of the two strings. Therefore, for Model 2, $\Pr(\mathbf{f}|\mathbf{e})$ does depend on the order of the words in $\mathbf{e}$ and $\mathbf{f}$. Although it is possible to obtain interesting correlations between some pairs of frequent words in the two languages using Models 1 and 2, as we will see later (in Figure 1.5), these models often lead to unsatisfactory word-by-word alignments of complete translations.

In Models 3, 4, and 5, we develop the French string by choosing for each word in the English string, first the number of words in the French string that will be connected to it, then the identity of these French words, and finally the actual positions in the French string that these words will occupy. It is this last step that determines the connections between the English string and the French string and it is here that these three models differ. In Model 3, as in Model 2, the probability of a connection depends on the positions that it connects and on the lengths of the English and French strings. In Model 4 the probability of a connection depends in addition on the identities of the French and English words connected and on the positions of any other French words that are connected to the same English word. Models 3 and 4 are deficient, a technical concept defined and discussed in Section 1.4.5. Briefly, this means that they waste some of their probability on objects that are not French strings at all. Model 5 is very much like Model 4, except that is not deficient.

Models 1 through 4 serve as stepping stones to the training of Model 5. Models 1 and 2 have an especially simple mathematical form so that iterations of the EM algorithm can be computed exactly. That is, we can explicitly perform sums over all possible alignments for these two models. In addition, Model 1 has a unique local maximum so that parameters derived for it in a series of EM iterations do not depend on the starting point for the iterations. As explained below, we use Model 1 to provide initial estimates for the parameters of Model 2. In Model 2 and subsequent models, the likelihood function does not have a unique local maximum, but by initializing each model from the parameters of the model before it, we arrive at estimates of the parameters of the final model that do not depend on our initial estimates of the parameters for Model 1.

In Models 3 and 4, we must be content with approximate EM iterations because it is not feasible to carry out sums over all possible alignments for these models. But, while approaching more closely the complexity of Model 5, they retain enough simplicity to allow an efficient investigation of the neighborhood of probable alignments and therefore allow us to include what we hope are all of the important alignments in each EM iteration.

In the remainder of this Section, we give an informal but reasonably precise description of each of the five models and an intuitive account of the EM algorithm as applied to them. We assume the reader to be comfortable with Lagrange multipliers, partial differentiation, and constrained optimization as they are presented in a typical college calculus text, and to have a nodding acquaintance with random variables. On his first time through, the reader may wish to jump from here directly to Section 1.5, returning to this Section when and if he should desire to understand more deeply how the results reported later are achieved.

The basic mathematical objects with which we deal here are the joint probability distribution $\Pr(\mathbf{F} = \mathbf{f}, \mathbf{A} = \mathbf{a}, \mathbf{E} = \mathbf{e})$ of the random variable $(\mathbf{F}, \mathbf{A}, \mathbf{E})$, where $\mathbf{F}$ is a random French string, $\mathbf{E}$ is a random English string, and $\mathbf{A}$ is a random alignment between them; and various marginal and conditional probability distributions that can be constructed from it, especially the distribution $\Pr(\mathbf{F} = \mathbf{f} | \mathbf{E} = \mathbf{e})$. We generally follow the common convention of using upper case letters to denote random variables and the corresponding lower case letters to denote specific values that the random variables may take. We have already used l and m to represent the lengths of the strings $\mathbf{e}$ and $\mathbf{f}$, and so we use L and M to denote the corresponding random variables. When there is no possibility for confusion, or, more properly, when the probability of confusion is not thereby materially increased, we write $\Pr(\mathbf{f}, \mathbf{a}, \mathbf{e})$ for $\Pr(\mathbf{F} = \mathbf{f}, \mathbf{A} = \mathbf{a}, \mathbf{E} = \mathbf{e})$, and use similar shorthands throughout.

We can write the likelihood of $(\mathbf{f}|\mathbf{e})$ in terms of the conditional probability $\Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$ as

$$\Pr(\mathbf{f}|\mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}). \tag{1.3}$$

The sum here,like all subsequent sums over $\mathbf{a}$, is over the elements of $\mathcal{A}(\mathbf{e}, \mathbf{f})$. We restrict ourselves in this Section to alignments like the one shown in Figure 1.1 where each French word has exactly one connection. In this kind of alignment, each cept is either a single English word or it is empty. Therefore, we can assign cepts to positions in the English string, reserving position zero for the empty cept. If the English string, $\mathbf{e} = e_1^l \equiv e_1 e_2 \ldots e_l$, has l words, and the French string, $\mathbf{f} = f_1^m \equiv f_1 f_2 \ldots f_m$, has m words, then the alignment, $\mathbf{a}$, can be represented by a series, $a_1^m \equiv a_1 a_2 \ldots a_m$, of m values, each between 0 and l such that if the word in position j of the French string is connected to the word in position i of the English string, then $a_j = i$, and if it is not connected to any English word, then $a_j = 0$.

Without loss of generality, we can write

$$\Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) \quad = \quad \Pr(m|\mathbf{e}) \prod_{j=1}^{m} \Pr(a_j | a_1^{j-1}, f_1^{j-1}, m, \mathbf{e}) \Pr(f_j | a_1^j, f_1^{j-1}, m, \mathbf{e}). \tag{1.4}$$

This is only one of many ways in which $\Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$ can be written as the product of a series of conditional probabilities. It is important to realize that Equation (1.4) is not an approximation. Regardless of the form of $\Pr(\mathbf{f}, \mathbf{a}|\mathbf{e})$, it can always be analyzed into a product of terms in this way. We are simply asserting in this equation that when we generate a French string together with an alignment from an English string, we can first choose the length of the French string given our knowledge of the English string. Then we can choose where to connect the first position in the French string given our knowledge of the English string and the length of the French string. Then we can choose the identity of the first word in the French string given our knowledge of the English string, the length of the French string, and the position in the English string to which the first position in the French string is connected, and so on. As we step through the French string, at each point we make our next choice given our complete knowledge of the English string and of all our previous choices as to the details of the French string and its alignment.

1.4.1 Model 1

The conditional probabilities on the right-hand side of Equation (1.4) cannot all be taken as independent parameters because there are too many of them. In Model 1, we assume that $\Pr(m|\mathbf{e})$ is independent of $\mathbf{e}$ and m; that $\Pr(a_j | a_1^{j-1}, f_1^{j-1}, m, \mathbf{e})$, depends only on l, the length of the English string, and therefore must be $(l+1)^{-1}$; and that $\Pr(f_j | a_1^j, f_1^{j-1}, m, \mathbf{e})$ depends only on f_j and e_{a_j}. The parameters, then, are $\epsilon \equiv \Pr(m|\mathbf{e})$, and $t(f_j | e_{a_j}) \equiv \Pr(f_j | a_1^j, f_1^{j-1}, m, \mathbf{e})$, which we call the *translation probability* of f_j given e_{a_j}. We think of ϵ as some small, fixed number. The distribution of M, the length of the French string, is unnormalized but this is a minor technical issue of no significance to our computations. If we wish, we can think of M as having some finite range. As long as this range encompasses everything that actually occurs in training data, no problems arise.

We turn now to the problem of estimating the translation probabilities for Model 1. The joint likelihood of a French string and an alignment given and English string is

$$\Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} t(f_j | e_{a_j}). \tag{1.5}$$

The alignment is determined by specifying the values of a_j for j from 1 to m, each of which can take any value from 0 to l. Therefore,

$$\Pr(\mathbf{f}|\mathbf{e}) = \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j | e_{a_j}). \tag{1.6}$$

We wish to adjust the translation probabilities so as to maximize $\Pr(\mathbf{f}|\mathbf{e})$ subject to the constraints that for each e,

$$\sum_f t(f|e) = 1. \tag{1.7}$$

Following standard practice for constrained maximization, we introduce Lagrange multipliers, λ_e, and seek an unconstrained extremum of the auxiliary function

$$h(t, \lambda) \equiv \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) - \sum_{e} \lambda_e (\sum_f t(f|e) - 1). \tag{1.8}$$

An extremum occurs when all of the partial derivatives of h with respect to the components of t and λ are zero. That the partial derivatives with respect to the components of λ be zero is simply a restatement of the constraints on the translation probabilities. The partial derivative of h with respect to $t(f|e)$ is

$$\frac{\partial h}{\partial t(f|e)} = \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \sum_{j=1}^{m} \delta(f, f_j)\delta(e, e_{a_j}) t(f|e)^{-1} \prod_{k=1}^{m} t(f_k|e_{a_k}) - \lambda_e, \tag{1.9}$$

where δ is the Kronecker delta function, equal to one when both of its arguments are the same and equal to zero otherwise. This partial derivative will be zero provided that

$$t(f|e) = \lambda_e^{-1} \frac{\epsilon}{(l+1)^m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \sum_{j=1}^{m} \delta(f, f_j)\delta(e, e_{a_j}) \prod_{k=1}^{m} t(f_k|e_{a_k}). \tag{1.10}$$

Superficially Equation (1.10) looks like a solution to the extremum problem, but it is not because the translation probabilities appear on both sides of the equal sign. Nonetheless, it suggests an iterative procedure for finding a solution: given an initial guess for the translation probabilities, we can evaluate the right-hand side of Equation (1.10) and use the result as a new estimate for $t(f|e)$. (Here and elsewhere, the Lagrange multipliers simply serve as a reminder that we need to normalize the translation probabilities so that they satisfy Equation (1.7).) This process, when applied repeatedly, is called the EM algorithm. That it converges to a stationary point of h in situations like this, as demonstrated in Appendix B, was first shown by Baum [15] and later by others [46].

With the aid of Equation (1.5), we can reexpress Equation (1.10) as

$$t(f|e) = \lambda_e^{-1} \sum_{\mathbf{a}} \Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) \underbrace{\sum_{j=1}^{m} \delta(f, f_j)\delta(e, e_{a_j})}_{\text{number of times } e \text{ connects to } f \text{ in } \mathbf{a}}. \tag{1.11}$$

We call the expected number of times that e connects to f in the translation $(\mathbf{f}|\mathbf{e})$ the *count* of f given e for $(\mathbf{f}|\mathbf{e})$ and denote it by $c(f|e; \mathbf{f}, \mathbf{e})$. By definition,

$$c(f|e; \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}) \sum_{j=1}^{m} \delta(f, f_j)\delta(e, e_{a_j}), \tag{1.12}$$

where $\Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}) = \Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) / \Pr(\mathbf{f}|\mathbf{e})$. If we replace λ_e by $\lambda_e \Pr(\mathbf{f}|\mathbf{e})$, then Equation (1.11) can be written very compactly as

$$t(f|e) = \lambda_e^{-1} c(f|e; \mathbf{f}, \mathbf{e}). \tag{1.13}$$

In practice, our training data consists of a set of translations, $(\mathbf{f}^{(1)}|\mathbf{e}^{(1)})$, $(\mathbf{f}^{(2)}|\mathbf{e}^{(2)})$, ..., $(\mathbf{f}^{(S)}|\mathbf{e}^{(S)})$, so this equation becomes

$$t(f|e) = \lambda_e^{-1} \sum_{s=1}^{S} c(f|e; \mathbf{f}^{(s)}, \mathbf{e}^{(s)}). \tag{1.14}$$

Here, λ_e serves only as a reminder that the translation probabilities must be normalized.

Usually, it is not feasible to evaluate the expectation in Equation (1.12) exactly. Even when we exclude multi-word cepts, there are still $(l+1)^m$ alignments possible for $(\mathbf{f}|\mathbf{e})$. Model 1, however, is special because by recasting Equation (1.6), we arrive at an expression that can be evaluated efficiently. The right-hand side of Equation (1.6) is a sum of terms each of which is a monomial in the translation probabilities. Each monomial contains m translation probabilities, one for each of the words in $\mathbf{f}$. Different monomials correspond to different ways of connecting words in $\mathbf{f}$ to cepts in $\mathbf{e}$ with every way appearing exactly once. By direct evaluation, we see that

$$\sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) = \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i). \tag{1.15}$$

An example may help to clarify this. Suppose that $m = 3$ and $l = 1$, and that we write t_{ji} as a shorthand for $t(f_j|e_i)$. Then the left-hand side of Equation (1.15) is $t_{10}\,t_{20}\,t_{30} + t_{10}\,t_{20}\,0t_{31} + \cdots + t_{11}\,t_{21}\,t_{30} + t_{11}\,t_{21}\,t_{31}$, and the left-hand side is $(t_{10}+t_{11})(t_{20}+t_{21})(t_{30}+t_{31})$. It is routine to verify that these are the same. Therefore, we can interchange the sums in Equation (1.6) with the product to obtain

$$\Pr(\mathbf{f}|\mathbf{e}) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j|e_i). \tag{1.16}$$

If we use this expression in place of Equation (1.6) when we write the auxiliary function in Equation (1.8), we find that

$$c(f|e;\mathbf{f},\mathbf{e}) = \frac{t(f|e)}{t(f|e_0)+\cdots+t(f|e_l)} \underbrace{\sum_{j=1}^{m} \delta(f,f_j)}_{\text{count of } f \text{ in } \mathbf{f}} \overbrace{\sum_{i=0}^{l} \delta(e,e_i)}^{\text{count of } e \text{ in } \mathbf{e}}. \tag{1.17}$$

Thus, the number of operations necessary to calculate a count is proportional to $l+m$ rather than to $(l+1)^m$ as Equation (1.12) might suggest.

The details of our initial guesses for $t(f|e)$ are unimportant because $\Pr(\mathbf{f}|\mathbf{e})$ has a unique local maximum for Model 1, as is shown in Appendix B. We choose all of the $t(f|e)$ to be equal, but any other choice that avoids zeros would lead to the same final solution.

1.4.2 Model 2

In Model 1, we take no cognizance of where words appear in either string. The first word in the French string is just as likely to be connected to a word at the end of the English string as to one at the beginning. In Model 2 we make the same assumptions as in Model 1 except that we assume that $\Pr(a_j|a_1^{j-1}, f_1^{j-1}, m, \mathbf{e})$ depends on j, a_j, and m, as well as on l. We introduce a set of *alignment probabilities,*

$$a(a_j|j,m,l) \equiv \Pr(a_j|a_1^{j-1}, f_1^{j-1}, m, l), \tag{1.18}$$

which satisfy the constraints

$$\sum_{i=0}^{l} a(i|j,m,l) = 1 \tag{1.19}$$

for each triple jml. In place of Equation (1.6), we have

$$\Pr(\mathbf{f}|\mathbf{e}) = \epsilon \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j|e_{a_j}) a(a_j|j,m,l). \tag{1.20}$$

Therefore, we seek an unconstrained extremum of the auxiliary function

$$h(t, a, \lambda, \mu) \equiv \epsilon \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j | e_{a_j}) a(a_j | j, m, l)$$
$$- \sum_e \lambda_e (\sum_f t(f|e) - 1) - \sum_j \mu_{jml} (\sum_i a(i|j, m, l) - 1) \quad (1.21)$$

The reader will easily verify that Equations (1.11), (1.13), and (1.14) carry over from Model 1 to Model 2 unchanged. We need a new count, $c(i|j, m, l; \mathbf{f}, \mathbf{e})$, the expected number of times that the word in position j of $\mathbf{f}$ is connected to the word in position i of $\mathbf{e}$. Clearly,

$$c(i|j, m, l; \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}) \delta(i, a_j). \quad (1.22)$$

In analogy with Equations (1.13) and (1.14), we have, for a single translation,

$$a(i|j, m, l) = \mu_{jml}^{-1} c(i|j, m, l; \mathbf{f}, \mathbf{e}), \quad (1.23)$$

and, for a set of translations,

$$a(i|j, m, l) = \mu_{jml}^{-1} \sum_{s=1}^{S} c(i|j, m, l; \mathbf{f}^{(s)}, \mathbf{e}^{(s)}). \quad (1.24)$$

Notice that if $\mathbf{f}^{(s)}$ does not have length m or if $\mathbf{e}^{(s)}$ does not have length l, then the corresponding count is zero. As with the λ's in earlier equations, the μ's here serve simply to remind us that the alignment probabilities must be normalized.

Model 2 shares with Model 1 the important property that the sums in Equations (1.12) and (1.22) can be obtained efficiently. We can rewrite Equation (1.20) as

$$\Pr(\mathbf{f}|\mathbf{e}) = \epsilon \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j | e_i) a(i|j, m, l). \quad (1.25)$$

Using this form for $\Pr(\mathbf{f}|\mathbf{e})$, we find that

$$c(f|e; \mathbf{f}, \mathbf{e}) = \sum_{j=1}^{m} \sum_{i=0}^{l} \frac{t(f|e)\, a(i|j, m, l)\, \delta(f, f_j) \delta(e, e_i)}{t(f|e_0)\, a(0|j, m, l) + \cdots + t(f|e_l)\, a(l|j, m, l)}, \quad (1.26)$$

and

$$c(i|j, m, l; \mathbf{f}, \mathbf{e}) = \frac{t(f_j | e_i)\, a(i|j, m, l)}{t(f_j | e_0)\, a(0|j, m, l) + \cdots + t(f_j | e_l)\, a(l|j, m, l)}. \quad (1.27)$$

Equation (1.26) has a double sum rather than the product of two single sums, as in Equation (1.17), because in Equation (1.26) i and j are tied together through the alignment probabilities.

Model 1 is the special case of Model 2 in which $a(i|j, m, l)$ is held fixed at $(l+1)^{-1}$. Therefore, any set of parameters for Model 1 can be reinterpreted as a set of parameters for Model 2. Taking as our initial estimates of the parameters for Model 2 the parameter values that result from training Model 1 is equivalent to computing the probabilities of all alignments as if we were dealing with Model 1, but then collecting the counts as if we were dealing with Model 2. The idea of computing the probabilities of the alignments using one model, but collecting the counts in a way appropriate to a second model is very general and can always be used to transfer a set of parameters from one model to another.

1.4.3 Intermodel interlude

We created Models 1 and 2 by making various assumptions to the conditional probabilities that appear in Equation (1.4). As we have mentioned, Equation (1.4) is an exact statement, but it is only one of many ways in which the joint likelihood of $\mathbf{f}$ and $\mathbf{a}$ can be written as a product of conditional probabilities. Each such product corresponds in a natural way to a generative process for developing $\mathbf{f}$ and $\mathbf{a}$ from $\mathbf{e}$. In the process corresponding to Equation (1.4), we first choose a length for $\mathbf{f}$. Next, we decide which position in $\mathbf{e}$ is connected to f_1 and what the identity of f_1 is. Then, we decide which position in $\mathbf{e}$ is connected to f_2, and so on. For Models 3, 4, and 5, we write the joint likelihood as a product of conditional probabilities in a different way.

Casual inspection of some translations quickly establishes that *the* is usually translated into a single word (*le, la,* or *l'*), but is sometimes omitted; or that *only* is often translated into one word (for example, *seulement*), but sometimes into two (for example, *ne ... que*), and sometimes into none. The number of French words to which e is connected in a randomly selected alignment is a random variable, Φ_e, that we call the *fertility* of e. Each choice of the parameters in Model 1 or Model 2 determines a distribution, $\Pr(\Phi_e = \phi)$, for this random variable. But the relationship is remote: just what change will be wrought in the distribution of Φ_{the} if, say, we adjust $a(1|2,8,9)$ is not immediately clear. In Models 3, 4, and 5, we parameterize fertilities directly.

As a prolegomenon to a detailed discussion of Models 3, 4, and 5, we describe the generative process upon which they are based. Given an English string, $\mathbf{e}$, we first decide the fertility of each word and a list of French words to connect to it. We call this list, which may be empty, a *tablet.* The collection of tablets is a random variable, T, that we call the *tableau* of $\mathbf{e}$; the tablet for the i^{th} English word is a random variable, T_i; and the k^{th} French word in the i^{th} tablet is a random variable, T_{ik}. After choosing the tableau, we permute its words to produce $\mathbf{f}$. This permutation is a random variable, Π. The position in $\mathbf{f}$ of the k^{th} word in the i^{th} tablet is yet another a random variable, Π_{ik}.

The joint likelihood for a tableau, τ, and a permutation, π, is

$$\begin{aligned}
\Pr(\tau,\pi|\mathbf{e}) \;=\; & \prod_{i=1}^{l} \Pr(\phi_i|\phi_1^{i-1},\mathbf{e}) \Pr(\phi_0|\phi_1^l,\mathbf{e}) \times \\
& \prod_{i=0}^{l}\prod_{k=1}^{\phi_i} \Pr(\tau_{ik}|\tau_{i1}^{k-1},\tau_0^{i-1},\phi_0^l,\mathbf{e}) \times \\
& \prod_{i=1}^{l}\prod_{k=1}^{\phi_i} \Pr(\pi_{ik}|\pi_{i1}^{k-1},\pi_1^{i-1},\tau_0^l,\phi_0^l,\mathbf{e}) \times \\
& \prod_{k=1}^{\phi_0} \Pr(\pi_{0k}|\pi_{01}^{k-1},\pi_1^l,\tau_0^l,\phi_0^l,\mathbf{e}). \qquad (1.28)
\end{aligned}$$

In this equation, τ_{i1}^{k-1} represents the series of values $\tau_{i1},\ldots,\tau_{ik-1}$; π_{i1}^{k-1} represents the series of values $\pi_{i1},\ldots,\pi_{ik-1}$; and ϕ_i is a shorthand for ϕ_{e_i}.

Knowing τ and π determines a French string and an alignment, but in general several different pairs τ,π may lead to the same pair $\mathbf{f},\mathbf{a}$. We denote the set of such pairs by $\langle\mathbf{f},\mathbf{a}\rangle$. Clearly, then

$$\Pr(\mathbf{f},\mathbf{a}|\mathbf{e}) = \sum_{(\tau,\pi)\in\langle\mathbf{f},\mathbf{a}\rangle} \Pr(\tau,\pi|\mathbf{e}). \qquad (1.29)$$

The number of elements in $\langle\mathbf{f},\mathbf{a}\rangle$ is $\prod_{i=0}^{l}\phi_i!$ because for each τ_i there are $\phi_i!$ arrangements that lead to the pair $\mathbf{f},\mathbf{a}$. Figure 1.4 shows the two tableaux for *(bon marché | cheap(1,2)).*

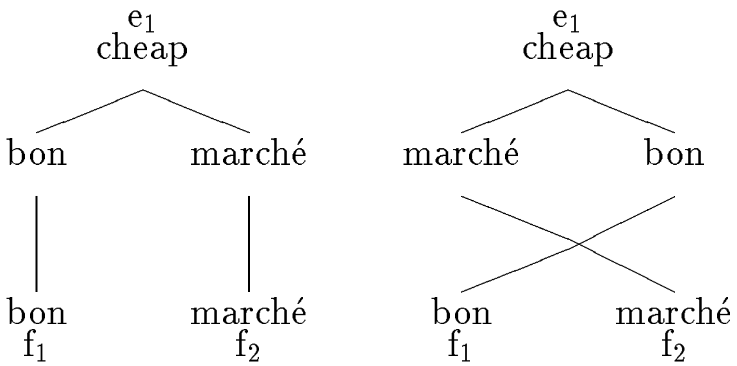

Figure 1.4: Two tableaux for one alignment.

Except for degenerate cases, there is one alignment in $\mathcal{A}(\mathbf{e},\mathbf{f})$ for which $\Pr(\mathbf{a}|\mathbf{e},\mathbf{f})$ is greatest. We call this the *Viterbi alignment* for $(\mathbf{f}|\mathbf{e})$ and denote it by $V(\mathbf{f}|\mathbf{e})$. We know of no practical algorithm for finding $V(\mathbf{f}|\mathbf{e})$ for a general model. Indeed, if someone were to claim that he had found $V(\mathbf{f}|\mathbf{e})$, we know of no practical algorithm for demonstrating that he is correct. But for Model 2 (and, thus, also for Model 1), finding $V(\mathbf{f}|\mathbf{e})$ is straightforward. For each j, we simply choose a_j so as to make the product $t(f_j|e_{a_j})a(a_j|j,m,l)$ as large as possible. The Viterbi alignment depends on the model with respect to which it is computed. When we need to distinguish between the Viterbi alignments for different models, we write $V(\mathbf{f}|\mathbf{e};1)$, $V(\mathbf{f}|\mathbf{e};2)$, and so on.

We denote by $\mathcal{A}_{i\leftarrow j}(\mathbf{e},\mathbf{f})$ the set of alignments for which $a_j = i$. We say that ij is pegged in these alignments. By the *pegged Viterbi alignment for ij*, which we write $V_{i\leftarrow j}(\mathbf{f}|\mathbf{e})$, we mean that element of $\mathcal{A}_{i\leftarrow j}(\mathbf{e},\mathbf{f})$ for which $\Pr(\mathbf{a}|\mathbf{e},\mathbf{f})$ is greatest. Obviously, we can find $V_{i\leftarrow j}(\mathbf{f}|\mathbf{e};1)$ and $V_{i\leftarrow j}(\mathbf{f}|\mathbf{e};2)$ quickly with a straightforward modification of the algorithm described above for finding $V(\mathbf{f}|\mathbf{e};1)$ and $V(\mathbf{f}|\mathbf{e};2)$

1.4.4 Model 3

Model 3 is based on Equation (1.28). Earlier, we were unable to treat each of the conditional probabilities on the right-hand side of Equation (1.4) as a separate parameter. With Equation (1.28) we are no better off and must again make assumptions to reduce the number of independent parameters. There are many different sets of assumptions that we might make, each leading to a different model for the translation process. In Model 3, we assume that, for i between 1 and l, $\Pr(\phi_i|\phi_1^{i-1},\mathbf{e})$ depends only on ϕ_i and e_i; that, for all i, $\Pr(\tau_{ik}|\tau_{i1}^{k-1},\tau_0^{i-1},\phi_0^l,\mathbf{e})$ depends only on τ_{ik} and e_i; and that, for i between 1 and l, $\Pr(\pi_{ik}|\pi_{i1}^{k-1},\pi_1^{i-1},\tau_0^l,\phi_0^l,\mathbf{e})$ depends only on π_{ik}, i, m, and l. The parameters of Model 3 are thus a set of *fertility probabilities*, $n(\phi|e_i) \equiv \Pr(\phi|\phi_1^{i-1},\mathbf{e})$; a set of translation probabilities, $t(f|e_i) \equiv \Pr(\mathrm{T}_{ik} = f|\tau_{i1}^{k-1},\tau_0^{i-1},\phi_0^l,\mathbf{e})$; and a set of *distortion probabilities*, $d(j|i,m,l) \equiv \Pr(\Pi_{ik} = j|\pi_{i1}^{k-1},\pi_1^{i-1},\tau_0^l,\phi_0^l,\mathbf{e})$.

We treat the distortion and fertility probabilities for e_0 differently. The empty cept conventionally occupies position 0, but actually has no position. Its purpose is to account for those words in the French string that cannot readily be accounted for by other cepts in the English string. Because we expect these words to be spread uniformly throughout the French string, and because they are placed only after all of the other words in the string have been placed, we assume that $\Pr(\Pi_{0k+1} = j|\pi_{01}^k,\pi_1^l,\tau_0^l,\phi_0^l,\mathbf{e})$ equals 0 unless position j is vacant in which case it equals $(\phi_0 - k)^{-1}$. Therefore, the contribution of the distortion probabilities for all of the words in τ_0 is $1/\phi_0!$.

We expect ϕ_0 to depend on the length of the French string because longer strings should have more extraneous words. Therefore, we assume that

$$\Pr(\phi_0|\phi_1^l, \mathbf{e}) = \binom{\phi_1 + \cdots + \phi_l}{\phi_0} p_0^{\phi_1+\cdots+\phi_l-\phi_0} p_1^{\phi_0} \tag{1.30}$$

for some pair of auxiliary parameters p_0 and p_1. The expression on the left-hand side of this equation depends on ϕ_1^l only through the sum $\phi_1 + \cdots + \phi_l$ and defines a probability distribution over ϕ_0 whenever p_0 and p_1 are nonnegative and sum to 1. We can interpret $\Pr(\phi_0|\phi_1^l, \mathbf{e})$ as follows. We imagine that each of the words from τ_1^l requires an extraneous word with probability p_1 and that this extraneous word must be connected to the empty cept. The probability that exactly ϕ_0 of the words from τ_1^l will require an extraneous word is just the expression given in Equation (1.30).

As with Models 1 and 2, an alignment of $(\mathbf{f}|\mathbf{e})$ is determined by specifying a_j for each position in the French string. The fertilities, ϕ_0 through ϕ_l, are functions of the a_j's: ϕ_i is equal to the number of j's for which a_j equals i. Therefore,

$$\begin{aligned}\Pr(\mathbf{f}|\mathbf{e}) &= \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \Pr(\mathbf{f}, \mathbf{a}|\mathbf{e}) \\ &= \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \binom{m-\phi_0}{\phi_0} p_0^{m-2\phi_0} p_1^{\phi_0} \prod_{i=1}^{l} \phi_i!\, n(\phi_i|e_i) \times \\ &\qquad \prod_{j=1}^{m} t(f_j|e_{a_j}) d(j|a_j, m, l) \end{aligned} \tag{1.31}$$

with $\sum_f t(f|e) = 1$, $\sum_j d(j|i, m, l) = 1$, $\sum_\phi n(\phi|e) = 1$, and $p_0 + p_1 = 1$. The assumptions that we make for Model 3 are such that each of the pairs (τ, π) in $\langle \mathbf{f}, \mathbf{a} \rangle$ makes an identical contribution to the sum in Equation(1.29). The factorials in Equation (1.31) come from carrying out this sum explicitly. There is no factorial for the empty cept because it is exactly cancelled by the contribution from the distortion probabilities.

By now, the reader will be able to provide his own auxiliary function for seeking a constrained minimum of the likelihood of a translation with Model 3, but for completeness and to establish notation, we write

$$\begin{aligned} h(t, d, n, p, \lambda, \mu, \nu, \xi) &= \Pr(\mathbf{f}|\mathbf{e}) - \sum_e \lambda_e (\textstyle\sum_f t(f|e) - 1) - \sum_i \mu_{iml} (\sum_j d(j|i, m, l) - 1) \\ &\qquad - \sum_e \nu_e (\textstyle\sum_\phi n(\phi|e) - 1) - \xi(p_0 + p_1 - 1). \end{aligned} \tag{1.32}$$

Following the trail blazed with Models 1 and 2, we define the counts

$$c(f|e; \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}) \sum_{j=1}^{m} \delta(f, f_j) \delta(e, e_{a_j}), \tag{1.33}$$

$$c(j|i, m, l; \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}) \delta(i, a_j), \tag{1.34}$$

$$c(\phi|e; \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}) \sum_{i=1}^{l} \delta(\phi, \phi_i) \delta(e, e_i), \tag{1.35}$$

$$c(0; \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{a}|\mathbf{e}, \mathbf{f})(m - 2\phi_0)$$

and

$$c(1;\mathbf{f},\mathbf{e}) = \sum_{\mathbf{a}} \Pr(\mathbf{a}|\mathbf{e},\mathbf{f})\phi_0. \tag{1.37}$$

The counts in these last two equations correspond to the parameters p_0 and p_1 that determine the fertility of the empty cept in the English string. The reestimation formulae for Model 3 are

$$t(f|e) = \lambda_e^{-1} \sum_{s=1}^{S} c(f|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)}), \tag{1.38}$$

$$d(j|i,m,l) = \mu_{iml}^{-1} \sum_{s=1}^{S} c(j|i,m,l;\mathbf{f}^{(s)},\mathbf{e}^{(s)}), \tag{1.39}$$

$$n(\phi|e) = \nu_e^{-1} \sum_{s=1}^{S} c(\phi|e;\mathbf{f}^{(s)},\mathbf{e}^{(s)}), \tag{1.40}$$

and

$$p_k = \xi^{-1} \sum_{s=1}^{S} c(k;\mathbf{f}^{(s)},\mathbf{e}^{(s)}). \tag{1.41}$$

Equations (1.33) and (1.38) are identical to Equations (1.12) and (1.14) and are repeated here only for convenience. Equations (1.34) and (1.39) are similar to Equations (1.22) and (1.24), but $a(i|j,m,l)$ differs from $d(j|i,m,l)$ in that the former sums to unity over all i for fixed j while the latter sums to unity over all j for fixed i. Equations (1.35), (1.36), (1.37), (1.40), and (1.41), for the fertility parameters, are new.

The trick that allows us to evaluate the right-hand sides of Equations (1.12) and (1.22) efficiently for Model 2 does not work for Model 3. Because of the fertility parameters, we cannot exchange the sums over a_1 through a_m with the product over j in Equation (1.31) as we were able to for Equations (1.6) and (1.20). We are not, however, entirely bereft of hope. The alignment is a useful device precisely because some alignments are much more probable than others. Our strategy is to carry out the sums in Equations (1.31) and (1.33) through (1.37) only over some of the more probable alignments, ignoring the vast sea of much less probable ones. Specifically, we begin with the most probable alignment that we can find and then include all alignments that can be obtained from it by small changes.

To define unambiguously the subset, $\mathcal{S}$, of the elements of $\mathcal{A}(\mathbf{f}|\mathbf{e})$ over which we evaluate the sums, we need yet more terminology. We say that two alignments, $\mathbf{a}$ and $\mathbf{a}'$, differ by a move if there is exactly one value of j for which $a_j \neq a_j'$. We say that they differ by a swap if $a_j = a_j'$ except at two values, j_1 and j_2, for which $a_{j_1} = a_{j_2}'$ and $a_{j_2} = a_{j_1}'$. We say that two alignments are *neighbors* if they are identical or differ by a move or by a swap. We denote the set of all neighbors of $\mathbf{a}$ by $\mathcal{N}(\mathbf{a})$.

Let $b(\mathbf{a})$ be that neighbor of $\mathbf{a}$ for which the likelihood $\Pr(b(\mathbf{a})|\mathbf{f},\mathbf{e})$ is greatest. Suppose that ij is pegged for $\mathbf{a}$. Among the neighbors of $\mathbf{a}$ for which ij is also pegged, let $b_{i\leftarrow j}(\mathbf{a})$ be that for which the likelihood is greatest. The sequence of alignments $\mathbf{a}$, $b(\mathbf{a})$, $b^2(\mathbf{a}) \equiv b(b(\mathbf{a}))$, ..., converges in a finite number of steps to an alignment that we write as $b^\infty(\mathbf{a})$. Similarly, if ij is pegged for $\mathbf{a}$, the sequence of alignments $\mathbf{a}$, $b_{i\leftarrow j}(\mathbf{a})$, $b_{i\leftarrow j}^2(\mathbf{a})$, ..., converges in a finite number of steps to an alignment that we write as $b_{i\leftarrow j}^\infty(\mathbf{a})$. The simple form of the distortion probabilities in Model 3 makes it easy to find $b(\mathbf{a})$ and $b_{i\leftarrow j}(\mathbf{a})$. If $\mathbf{a}'$ is a neighbor of $\mathbf{a}$ obtained from it by the move of j from i to i', and if neither i nor i' is 0, then

$$\Pr(\mathbf{a}'|\mathbf{e},\mathbf{f}) = \Pr(\mathbf{a}|\mathbf{e},\mathbf{f}) \frac{(\phi_{i'}+1)}{\phi_i} \frac{n(\phi_{i'}+1|e_{i'})}{n(\phi_{i'}|e_{i'})} \frac{n(\phi_i-1|e_i)}{n(\phi_i|e_i)} \frac{t(f_j|e_{i'})}{t(f_j|e_i)} \frac{d(j|i',m,l)}{d(j|i,m,l)}. \tag{1.42}$$

Notice that $\phi_{i'}$ is the fertility of the word in position i' for alignment $\mathbf{a}$. The fertility of this word in alignment $\mathbf{a}'$ is $\phi_{i'} + 1$. Similar equations can be easily derived when either i or i' is zero, or when $\mathbf{a}$ and $\mathbf{a}'$ differ by a swap. We leave the details to the reader.

With these preliminaries, we define $\mathcal{S}$ by

$$\mathcal{S} = \mathcal{N}(b^{\infty}(V(\mathbf{f}|\mathbf{e};2))) \bigcup \bigcup_{ij} \mathcal{N}(b^{\infty}_{i\leftarrow j}(V_{i\leftarrow j}(\mathbf{f}|\mathbf{e};2))). \tag{1.43}$$

In this equation, we use $b^{\infty}(V(\mathbf{f}|\mathbf{e};2))$ and $b^{\infty}_{i\leftarrow j}(V_{i\leftarrow j}(\mathbf{f}|\mathbf{e};2))$ as handy approximations to $V(\mathbf{f}|\mathbf{e};3)$ and $V_{i\leftarrow j}(\mathbf{f}|\mathbf{e};3)$ neither of which we are able to compute efficiently.

In one iteration of the EM algorithm for Model 3, we compute the counts in Equations (1.33) through (1.37), summing only over elements of $\mathcal{S}$, and then use these counts in Equations (1.38) through (1.41) to obtain a new set of parameters. If the error made by including only some of the elements of $\mathcal{A}(\mathbf{e}, \mathbf{f})$ is not too great, this iteration will lead to values of the parameters for which the likelihood of the training data is at least as large as for the first set of parameters.

We make no initial guess of the parameters for Model 3, but instead adapt the parameters from the final iteration of the EM algorithm for Model 2. That is, we compute the counts in Equations (1.33) through (1.37) using Model 2 to evaluate $\Pr(\mathbf{a}|\mathbf{e}, \mathbf{f})$. The simple form of Model 2 again makes exact calculation feasible. We can readily adapt Equations (1.26) and (1.27) to compute counts for the translation and distortion probabilities; efficient calculation of the fertility counts is more involved, and we defer a discussion of it to Appendix B.

1.4.5 Deficiency

The reader will have noticed a problem with our parameterization of the distortion probabilities in Model 3: whereas we can see by inspection that the sum over all pairs τ, π of the expression on the right-hand side of Equation (1.28) is unity, it is equally clear that this can no longer be the case if we assume that $\Pr(\Pi_{ik} = j|\pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^l, \phi_0^l, \mathbf{e})$ depends only on j, i, m, and l for $i > 0$. Because the distortion probabilities for assigning positions to later words do not depend on the positions assigned to earlier words, Model 3 wastes some of its probability on what we might call generalized strings, *i.e.,* strings that have some positions with several words and others with none. When a model has this property of not concentrating all of its probability on events of interest, we say that it is *deficient.* Deficiency is the price that we pay for the simplicity that allows us to write Equation (1.42).

Deficiency poses no serious problem here. Although Models 1 and 2 are not technically deficient, they are surely spiritually deficient. Each assigns the same probability to the alignments *(Je n'ai pas de stylo | I(1) do not(2,4) have(3) a(5) pen(6))* and *(Je pas ai ne de stylo | I(1) do not(2,4) have(3) a(5) pen(6)),* and, therefore, essentially the same probability to the translations *(Je n'ai pas de stylo | I do not have a pen)* and *(Je pas ai ne de stylo | I do not have a pen).* In each case, *not* produces two words, *ne* and *pas,* and in each case, one of these words ends up in the second position of the French string and the other in the fourth position. The first translation should be much more probable than the second, but this defect is of little concern because while we might have to translate the first string someday, we will never have to translate the second. We do not use our translation models to predict French given English but rather as a component of a system designed to predict English given French. They need only be accurate to within a constant factor over well-formed strings of French words.

1.4.6 Model 4

Often the words in an English string constitute phrases that are translated as units into French. Sometimes, a translated phrase may appear at a spot in the French string different from that at which the corresponding English phrase appears in the English string. The distortion probabilities of Model 3 do not account well for this tendency of phrases to move around as units. Movement of a long phrase will be much less likely than movement of a short phrase because each word must be moved independently. In Model 4, we modify our treatment of $\Pr(\Pi_{ik} = j|\pi_{i1}^{k-1}, \pi_1^{i-1}, \tau_0^l, \phi_0^l, \mathbf{e})$ so as to alleviate this problem. Words that are connected to the empty cept do not usually form phrases and so we continue to assume that these words are spread uniformly throughout the French string.

As we have described, an alignment resolves an English string into a ceptual scheme consisting of a set of possibly overlapping cepts. Each of these cepts then accounts for one or more French words. In Model 3 the ceptual scheme for an alignment is determined by the fertilities of the words: a word is a cept if its fertility is greater than zero. The empty cept is a part of the ceptual scheme if ϕ_0 is greater than zero. As before we exclude multi-word cepts. Among the one-word cepts, there is a natural order corresponding to the order in which they appear in the English string. Let $[i]$ denote the position in the English string of the i^{th} one-word cept. We define the *center* of this cept, $\odot_i$, to be the ceiling of the average value of the positions in the French string of the words from its tablet. We define its *head* to be that word in its tablet for which the position in the French string is smallest.

In Model 4, we replace $d(j|i, m, l)$ by two sets of parameters: one for placing the head of each cept, and one for placing any remaining words. For $[i] > 0$, we require that the head for cept i be $\tau_{[i]1}$ and we assume that

$$\Pr(\Pi_{[i]1} = j|\pi_1^{[i]-1}, \tau_0^l, \phi_0^l, \mathbf{e}) = d_1(j - \odot_{i-1}|\mathcal{A}(e_{[i-1]}), \mathcal{B}(f_j)). \tag{1.44}$$

Here, $\mathcal{A}$ and $\mathcal{B}$ are functions of the English and French word that take on a small number of different values as their arguments range over their respective vocabularies. Brown *et al.* [34] describe an algorithm for dividing a vocabulary into classes so as to preserve mutual information between adjacent classes in running text. We construct $\mathcal{A}$ and $\mathcal{B}$ as functions with fifty distinct values by dividing the English and French vocabularies each into fifty classes according to this algorithm. By assuming that the probability depends on the previous cept and on the identity of the French word being placed, we can account for such facts as the appearance of adjectives before nouns in English but after them in French. We call $j - \odot_{i-1}$ the displacement for the head of cept i. It may be either positive or negative. We expect $d_1(-1|\mathcal{A}(e), \mathcal{B}(f))$ to be larger than $d_1(+1|\mathcal{A}(e), \mathcal{B}(f))$ when e is an adjective and f is a noun. Indeed, this is borne out in the trained distortion probabilities for Model 4, where we find that $d_1(-1|\mathcal{A}(\textit{government's}), \mathcal{B}(\textit{développement}))$ is 0.7986, while $d_1(+1|\mathcal{A}(\textit{government's}), \mathcal{B}(\textit{développement}))$ is 0.0168.

Suppose, now, that we wish to place the k^{th} word of cept i for $[i] > 0$, $k > 1$. We assume that

$$\Pr(\Pi_{[i]k} = j|\pi_{[i]1}^{k-1}, \pi_1^{[i]-1}, \tau_0^l, \phi_0^l, \mathbf{e}) = d_{>1}(j - \pi_{[i]k-1}|\mathcal{B}(f_j)). \tag{1.45}$$

We require that $\pi_{[i]k}$ be greater than $\pi_{[i]k-1}$. Some English words tend to produce a series of French words that belong together, while others tend to produce a series of words that should be separate. For example, *implemented* can produce *mis en application,* which usually occurs as a unit, but *not* can produce *ne pas,* which often occurs with an intervening verb. We expect $d_{>1}(2|\mathcal{B}(\textit{pas}))$ to be relatively large compared with $d_{>1}(2|\mathcal{B}(\textit{en}))$. After training, we find that $d_{>1}(2|\mathcal{B}(\textit{pas}))$ is 0.6847 and $d_{>1}(2|\mathcal{B}(\textit{en}))$ is 0.1533.

Whereas we assume that $\tau_{[i]1}$ can be placed either before or after any previously positioned words, we require subsequent words from $\tau_{[i]}$ to be placed in order. This does not mean that they must occupy consecutive positions but only that the second word from $\tau_{[i]}$ must lie to the right of the first, the third to the right of the second, and so on. Because of this, only one of the $\phi_{[i]}!$ arrangements of $\tau_{[i]}$ is possible.

We leave the routine details of deriving the count and reestimation formulae for Model 4 to the reader. He may find the general formulae in Appendix B helpful. Once again, the several counts for a translation are expectations of various quantities over the possible alignments with the probability of each alignment computed from an earlier estimate of the parameters. As with Model 3, we know of no trick for evaluating these expectations and must rely on sampling some small set, $\mathcal{S}$, of alignments. As described above, the simple form that we assume for the distortion probabilities in Model 3, makes it possible for us to find $b^{\infty}(\mathbf{a})$ rapidly for any $\mathbf{a}$. The analogue of Equation (1.42) for Model 4 is complicated by the fact that when we move a French word from cept to cept we change the centers of two cepts and may affect the contribution of several words. It is nonetheless possible to evaluate the adjusted likelihood incrementally, although it is substantially more time consuming.

Faced with this unpleasant situation, we proceed as follows. Let the neighbors of $\mathbf{a}$ be ranked so that the first is the neighbor for which $\Pr(\mathbf{a}|\mathbf{e},\mathbf{f};3)$ is greatest, the second the one for which $\Pr(\mathbf{a}|\mathbf{e},\mathbf{f};3)$ is next greatest, and so on. We define $\tilde{b}(\mathbf{a})$ to be the highest ranking neighbor of $\mathbf{a}$ for which $\Pr(\tilde{b}(\mathbf{a})|\mathbf{e},\mathbf{f};4)$ is at least as large as $\Pr(\mathbf{a}|\mathbf{e},\mathbf{f};4)$. We define $\tilde{b}_{i\leftarrow j}(\mathbf{a})$ analogously. Here, $\Pr(\mathbf{a}|\mathbf{e},\mathbf{f};3)$ means $\Pr(\mathbf{a}|\mathbf{e},\mathbf{f})$ as computed with Model 3, and $\Pr(\mathbf{a}|\mathbf{e},\mathbf{f};4)$ means $\Pr(\mathbf{a}|\mathbf{e},\mathbf{f})$ as computed with Model 4. We define $\mathcal{S}$ for Model 4 by

$$\mathcal{S} = \mathcal{N}(\tilde{b}^{\infty}(V(\mathbf{f}|\mathbf{e};2))) \bigcup \bigcup_{ij} \mathcal{N}(\tilde{b}^{\infty}_{i\leftarrow j}(V_{i\leftarrow j}(\mathbf{f}|\mathbf{e};2))). \tag{1.46}$$

This equation is identical to Equation (1.46) except that b has been replaced by $\tilde{b}$.

1.4.7 Model 5

Models 3 and 4 are both deficient. In Model 4, not only can several words lie on top of one another, but words can be placed before the first position or beyond the last position in the French string. We remove this deficiency in Model 5.

After we have placed the words for $\tau_1^{[i]-1}$ and $\tau_{[i]1}^{\;k-1}$ there will remain some vacant positions in the French string. Obviously, $\tau_{[i]k}$ should be placed in one of these vacancies. Models 3 and 4 are deficient precisely because we fail to enforce this constraint for the one-word cepts. Let $v(j, \tau_1^{[i]-1}, \tau_{[i]1}^{\;k-1})$ be the number of vacancies up to and including position j just before we place $\tau_{[i]k}$. In the interest of notational brevity, a noble but elusive goal, we write this simply as v_j. We retain two sets of distortion parameters, as in Model 4, and continue to refer to them as d_1 and $d_{>1}$. We assume that, for $[i] > 0$,

$$\Pr(\Pi_{[i]1} = j|\pi_1^{[i]-1}, \tau_0^l, \phi_0^l, \mathbf{e}) = d_1(v_j|\mathcal{B}(f_j), v_{\odot_{i-1}}, v_m - \phi_{[i]} + 1)(1 - \delta(v_j, v_{j-1})). \tag{1.47}$$

The number of vacancies up to j is the same as the number of vacancies up to $j-1$ only when j is not, itself, vacant. The last factor, therefore, is 1 when j is vacant and 0 otherwise. In the final parameter of d_1, v_m is the number of vacancies remaining in the French string. If $\phi_{[i]} = 1$, then $\tau_{[i]1}$ may be placed in any of these vacancies; if $\phi_{[i]} = 2$, $\tau_{[i]1}$ may be placed in any but the last of these vacancies; in general, $\tau_{[i]1}$ may be placed in any but the rightmost $\phi_{[i]} - 1$ of the remaining vancancies. Because $\tau_{[i]1}$ must occupy the leftmost place of any of the words from $\mathrm{T}_{[i]}$, we must take care to leave room at the end of the string for the remaining words from this tablet. As with Model 4, we allow d_1 to depend on the center of the previous cept and on f_j, but we suppress the dependence on $e_{[i-1]}$ since we should otherwise have too many parameters.

For $[i] > 0$ and $k > 1$, we assume

$$\Pr(\Pi_{[i]k} = j|\pi_{[i]1}^{\;k-1}, \pi_1^{[i]-1}, \tau_0^l, \phi_0^l, \mathbf{e}) = d_{>1}(v_j - v_{\pi_{[i]k-1}}|\mathcal{B}(f_j), v_m - v_{\pi_{[i]k-1}} - \phi_{[i]} + k)(1 - \delta(v_j, v_{j-1})). \tag{1.48}$$

Again, the final factor enforces the constraint that $\tau_{[i]k}$ land in a vacant position, and, again, we assume that the probability depends on f_j only through its class. Model 5 is described in more detail in Appendix B.

As with Model 4, we leave the details of the count and reestimation formulae to the reader. No incremental evaluation of the likelihood of neighbors is possible with Model 5 because a move or swap may require wholesale recomputation of the likelihood of an alignment. Therefore, when we evaluate expectations for Model 5, we include only the alignments in $\mathcal{S}$ as defined in Equation (1.46). We further trim these alignments by removing any alignment, $\mathbf{a}$, for which $\Pr(\mathbf{a}|\mathbf{e}, \mathbf{f}; 4)$ is too much smaller than $\Pr(\tilde{b}^{\infty}(V(\mathbf{f}|\mathbf{e}; 2)|\mathbf{e}, \mathbf{f}; 4)$.

Model 5 is a powerful but unwieldy ally in the battle to align translations. It must be led to the battlefield by its weaker but more agile brethren Models 2, 3, and 4. In fact, this is the *raison d'être* of these models. To keep them aware of the lay of the land, we adjust their parameters as we carry out iterations of the EM algorithm for Model 5. That is, we collect counts for Models 2, 3, and 4 by summing over alignments as determined by the abbreviated $\mathcal{S}$ described above, using Model 5 to compute $\Pr(\mathbf{a}|\mathbf{e}, \mathbf{f})$. Although this appears to increase the storage necessary for maintaining counts as we proceed through the training data, the extra burden is small because the overwhelming majority of the storage is devoted to counts for $t(f|e)$, and these are the same for Models 2, 3, 4, and 5.

1.5 Results

We have used a large collection of training data to estimate the parameters of the models described above. Brown, Lai, and Mercer, [28] have described an algorithm with which one can reliably extract French and English sentences that are translations of one another from parallel corpora. They used the algorithm to extract a large number of translations from several years of the proceedings of the Canadian parliament. From these translations, we have chosen as our training data those for which both the English sentence and the French sentence are thirty words or less in length. This is a collection of 1,778,620 translations. In an effort to eliminate some of the typographical errors that abound in the text, we have chosen as our English vocabulary all of those words that appear at least twice in English sentences in our data, and as our French vocabulary all of those words that appear at least twice in French sentences in our data. All other words we replace with a special *unknown English word* or *unknown French word* according as they appear in an English sentence or a French sentence. We arrive in this way at an English vocabulary of 42,005 words and a French vocabulary of 58,016 words. Some typographical errors are quite frequent, for example, *momento* for *memento,* and so our vocabularies are not completely free of them. At the same time, some words are truly rare, and so we have, in some cases, snubbed legitimate words. Adding e_0 to the English vocabulary brings it to $42,006$ words.

We have carried out twelve iterations of the EM algorithm for this data. We initialized the process by setting each of the $2,437,020,096$ translation probabilities, $t(f|e)$, to 1/58016. That is, we assume each of the $58,016$ words in the French vocabulary to be equally likely as a translation for each of the $42,006$ words in the English vocabulary. For $t(f|e)$ to be greater than zero at the maximum likelihood solution for one of our models, f and e must occur together in at least one of the translations in our training data. This is the case for only $25,427,016$ pairs, or about one percent of all translation probabilities. On the average, then, each English word appears with about 605 French words.

Table 1.1 summarizes our training computation. At each iteration, we compute the probabilities of the various alignments of each translation using one model, and collect counts using a second (possibly different) model. These are referred to in the table as the In model and the Out model, respectively. After each iteration, we retain individual values only for those translation

Iteration	In	$\rightarrow$	Out	Survivors	Alignments	Perplexity
1	1	$\rightarrow$	2	12,017,609		71,550.56
2	2	$\rightarrow$	2	12,160,475		202.99
3	2	$\rightarrow$	2	9,403,220		89.41
4	2	$\rightarrow$	2	6,837,172		61.59
5	2	$\rightarrow$	2	5,303,312		49.77
6	2	$\rightarrow$	2	4,397,172		46.36
7	2	$\rightarrow$	3	3,841,470		45.15
8	3	$\rightarrow$	5	2,057,033	291	124.28
9	5	$\rightarrow$	5	1,850,665	95	39.17
10	5	$\rightarrow$	5	1,763,665	48	32.91
11	5	$\rightarrow$	5	1,703,393	39	31.29
12	5	$\rightarrow$	5	1,658,364	33	30.65

Table 1.1: A summary of the training iterations.

probabilities that surpass a threshold; the remainder we set to a small value (10^{-12}). This value is so small that it does not affect the normalization conditions, but is large enough that translation probabilities can be resurrected during later iterations. We see in columns 4 and 5 that even though we lower the threshold as iterations progress, fewer and fewer probabilities survive. By the final iteration, only $1,658,364$ probabilities survive, an average of about thirty-nine French words for each English word.

Although the entire t array has $2,437,020,096$ entries, and we need to store it twice, once as probabilities and once as counts, it is clear from the preceeding remarks that we need never deal with more than about 25 million counts or about 12 million probabilities. We store these two arrays using standard sparse matrix techniques. We keep counts as single bytes, but allow for an overflow into two bytes if necessary. In this way, it is possible to run the training program in less than 100 megabytes of memory. While this number would have seemed extravagant a few years ago, today it is available at modest cost in a personal workstation.

As we have described, when the In model is neither Model 1 nor Model 2, we evaluate the count sums over only some of the possible alignments. Many of these alignments have a probability much smaller than that of the Viterbi alignment. The column headed *Alignments* in Table 1.1 shows the average number of alignments for which the probability is within a factor of 25 of the probability of the Viterbi alignment in each iteration. As this number drops, the model concentrates more and more probability onto fewer and fewer alignments so that the Viterbi alignment becomes ever more dominant.

The last column in the table shows the perplexity of the French text given the English text for the In model of the iteration. We expect the likelihood of the training data to increase with each iteration. We can think of this likelihood as arising from a product of factors, one for each French word in the training data. We have $28,850,104$ French words in our training data so the $28,850,104^{\text{th}}$ root of the likelihood is the average factor by which the likelihood is reduced for each additional French word. The reciprocal of this root is the perplexity shown in the table. As the likelihood increases, the perplexity decreases. We see a steady decrease in perplexity as the iterations progress except when we switch from Model 2 as the In model to Model 3. This sudden jump is not because Model 3 is a poorer model than Model 2, but because Model 3 is deficient: the great majority of its probability is squandered on objects that are not strings of French words. As we have argued, deficiency is not a problem. In our description of Model 1, we left $\Pr(m|\mathbf{e})$ unspecified. In quoting perplexities for Models 1 and 2, we have assumed that the length of the French string is Poisson with a mean that is a linear function of the length of the English string. Specifically, we have assumed that $\Pr(M = m|\mathbf{e}) = (\lambda l)^m e^{-\lambda l}/m!$, with λ equal to 1.09.

It is interesting to see how the Viterbi alignments change as the iterations progress. In Figure 1.5, we show for several sentences the Viterbi alignment after iterations 1, 6, 7, and 12. Iteration 1 is the first iteration for Model 2, and iterations 6, 7, and 12 are the final iterations for Models 2, 3, and 5, respectively. In each example, we show the French sentence with a subscript affixed to each word to ease the reader's task in interpreting the list of numbers after each English word. In the first example, *(Il me semble faire signe que oui | It seems to me that he is nodding),* two interesting changes evolve over the course of the iterations. In the alignment for Model 1, *Il* is correctly connected to *he,* but in all later alignments *Il* is incorrectly connected to *It.* Models 2, 3, and 5 discount a connection of *he* to *Il* because it is quite far away. We do not yet have a model with sufficient linguistic sophistication to make this connection properly. On the other hand, we see that *nodding,* which in Models 1, 2, and 3 is connected only to *signe* and *oui,* is correctly connected to the entire phrase *faire signe que oui* in Model 5. In the second example, *(Voyez les profits que ils ont réalisés | Look at the profits they have made),* Models 1, 2, and 3 incorrectly connect $\textit{profits}_4$ to both $\textit{profits}_3$ and $\textit{réalisés}_7$, but with Model 5, $\textit{profits}_4$ is correctly connected only to $\textit{profits}_3$ and $\textit{made}_7$ is connected to $\textit{réalisés}_7$. Finally, in *(De les promesses, de les promesses! | Promises, promises.),* $\textit{Promises}_1$ is connected to both instances of *promesses* with Model 1; $\textit{promises}_3$ is connected to most of the French sentence with Model 2; the final punctuation of the English sentence is connected to both the exclamation point and, curiously, to $\textit{de}_5$ with Model 3; and only with Model 5 do we have a satisfying alignment of the two sentences. The orthography for the French sentence in the second example is *Voyez les profits qu'ils ont réalisés* and in the third example is *Des promesses, des promesses!* We have restored the *e* to the end of *qu'* and have twice analyzed *des* into its constituents, *de* and *les.* We commit these and other petty pseudographic improprieties in the interest of regularizing the French text. In all cases, orthographic French can be recovered by rule from our corrupted versions.

Figures 1.6 through 1.15 show the translation probabilities and fertilities after the final iteration of training for a number of English words. We show all and only those probabilities that are greater than 0.01. Some words, like *nodding,* in Figure 1.6, do not slip gracefully into French. Thus, we have translations like *(Il fait signe que oui | He is nodding), (Il fait un signe de la tête | He is nodding), (Il fait un signe de tête affirmatif | He is nodding),* or *(Il hoche la tête affirmativement | He is nodding).* As a result, *nodding* frequently has a large fertility and spreads its translation probability over a variety of words. In French, what is worth saying is worth saying in many different ways. We see another facet of this with words like *should,* in Figure 1.7, which rarely has a fertility greater than one but still produces many different words, among them *devrait, devraient, devrions, doit, doivent, devons,* and *devrais.* These are (just a fraction of the many) forms of the French verb *devoir.* Adjectives fare a little better: *national,* in Figure 1.8, almost never produces more than one word and confines itself to one of *nationale, national, nationaux,* and *nationales,* respectively the feminine, the masculine, the masculine plural, and the feminine plural of the corresponding French adjective. It is clear that our models would benefit from some kind of morphological processing to rein in the lexical exuberance of French.

We see from the data for *the,* in Figure 1.9, that it produces *le, la, les,* and *l'* as we would expect. Its fertility is usually 1, but in some situations English prefers an article where French does not and so about 14% of the time its fertility is 0. Sometimes, as with *farmers,* in Figure 1.10, it is French that prefers the article. When this happens, the English noun trains to

Il_1 me_2 $semble_3$ $faire_4$ $signe_5$ que_6 oui_7

It seems(3) to(4) me(2) that(6) he(1) is nodding(5,7)
It(1) seems(3) to me(2) that he is nodding(5,7)
It(1) seems(3) to(4) me(2) that(6) he is nodding(5,7)
It(1) seems(3) to me(2) that he is nodding(4,5,6,7)

$Voyez_1$ les_2 $profits_3$ que_4 ils_5 ont_6 réalisés_7

Look(1) at the(2) profits(3,7) they(5) have(6) made
Look(1) at the(2,4) profits(3,7) they(5) have(6) made
Look(1) at the profits(3,7) they(5) have(6) made
Look(1) at the(2) profits(3) they(5) have(6) made(7)

De_1 les_2 $promesses_3$ $,_4$ de_5 les_6 $promesses_7$ $!_8$

Promises(3,7) ,(4) promises .(8)
Promises ,(4) promises(2,3,6,7) .(8)
Promises(3) ,(4) promises(7) .(5,8)
Promises(2,3) ,(4) promises(6,7) .(8)

Figure 1.5: The progress of alignments with iteration.

nodding

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
signe	0.164	4	0.342
la	0.123	3	0.293
tête	0.097	2	0.167
oui	0.086	1	0.163
fait	0.073	0	0.023
que	0.073		
hoche	0.054		
hocher	0.048		
faire	0.030		
me	0.024		
approuve	0.019		
qui	0.019		
un	0.012		
faites	0.011		

Figure 1.6: Translation and fertility probabilities for *nodding.*

produce its translation together with an article. Thus, *farmers* typically has a fertility 2 and usually produces either *agriculteurs* or *les.* We include additional examples in Figures 1.11 through 1.15, which show the translation and fertility probabilities for *external, answer, oil, former,* and *not.* Although we show the various probabilities to three decimal places, one must realize that the specific numbers that appear are peculiar to the training data that we used in obtaining them. They are not constants of nature relating the Platonic ideals of eternal English and eternal French. Had we used different sentences as training data, we might well have arrived at different numbers. For example, in Figure 1.9, we see that $t(le|the) = 0.497$ while the corresponding number from Figure 4 of Brown *et al.* [27] is 0.610. The difference arises not from some instability in the training algorithms or some subtle shift in the languages in recent years, but from the fact that we have used 1,778,620 pairs of sentences covering virtually the complete vocabulary of the Hansard data for training while they used only 40,000 pairs of sentences and restricted their attention to the 9,000 most common words in each of the two vocabularies.

Figures 1.16, 1.17, and 1.18 show automatically derived alignments for three translations. In the terminology of Section 1.4.6, each alignment is $\tilde{b}^{\infty}(V(\mathbf{f}|\mathbf{e};2))$. We stress that these alignments have been found by an algorithm that involves no explicit knowledge of either French or English. Every fact adduced to support them has been discovered algorithmically from the $1,778,620$ translations that constitute our training data. This data, in turn, is the product of an algorithm the sole linguistic input of which is a set of rules explaining how to find sentence boundaries in the two languages. We may justifiably claim, therefore, that these alignments are inherent in the Canadian Hansard data itself.

In the alignment shown in Figure 1.16, all but one of the English words has fertility 1. The final prepositional phrase has been moved to the front of the French sentence, but otherwise the translation is almost verbatim. Notice, however, that *the new proposal* has been translated into *les nouvelles propositions,* demonstrating that number is not an invariant under translation. The empty cept has fertility 5 here. It generates en_1, de_3, the comma, de_{16}, and de_{18}.

In Figure 1.17, two of the English words have fertility 0, one has fertility 2, and one, *embattled,* has fertility 5. *Embattled* is another word, like *nodding,* that eludes the French grasp and comes *should*

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
devrait	0.330	1	0.649
devraient	0.123	0	0.336
devrions	0.109	2	0.014
faudrait	0.073		
faut	0.058		
doit	0.058		
aurait	0.041		
doivent	0.024		
devons	0.017		
devrais	0.013		

Figure 1.7: Translation and fertility probabilities for *should.*

national

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
nationale	0.469	1	0.905
national	0.418	0	0.094
nationaux	0.054		
nationales	0.029		

Figure 1.8: Translation and fertility probabilities for *national.*

the

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
le	0.497	1	0.746
la	0.207	0	0.254
les	0.155		
l'	0.086		
ce	0.018		
cette	0.011		

Figure 1.9: Translation and fertility probabilities for *the.*

farmers

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
agriculteurs	0.442	2	0.731
les	0.418	1	0.228
cultivateurs	0.046	0	0.039
producteurs	0.021		

Figure 1.10: Translation and fertility probabilities for *farmers.*

external

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
extérieures	0.944	1	0.967
extérieur	0.015	0	0.028
externe	0.011		
extérieurs	0.010		

Figure 1.11: Translation and fertility probabilities for *external.*

answer

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
réponse	0.442	1	0.809
répondre	0.233	2	0.115
répondu	0.041	0	0.074
à	0.038		
solution	0.027		
répondez	0.021		
répondrai	0.016		
réponde	0.014		
y	0.013		
ma	0.010		

Figure 1.12: Translation and fertility probabilities for *answer.*

oil

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
pétrole	0.558	1	0.760
pétrolières	0.138	0	0.181
pétrolière	0.109	2	0.057
le	0.054		
pétrolier	0.030		
pétroliers	0.024		
huile	0.020		
Oil	0.013		

Figure 1.13: Translation and fertility probabilities for *oil.*

former

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
ancien	0.592	1	0.866
anciens	0.092	0	0.074
ex	0.092	2	0.060
précédent	0.054		
l'	0.043		
ancienne	0.018		
été	0.013		

Figure 1.14: Translation and fertility probabilities for *former.*

not

f	$t(f \mid e)$	ϕ	$n(\phi \mid e)$
ne	0.497	2	0.735
pas	0.442	0	0.154
non	0.029	1	0.107
rien	0.011		

Figure 1.15: Translation and fertility probabilities for *not.*

with a panoply of multi-word translations.

The final example, in Figure 1.18, has several features that bear comment. The second word, *Speaker,* is connected to the sequence *l'Orateur.* Like *farmers* above, it has trained to produce both the word that we naturally think of as its translation and the associated article. In our data, *Speaker* always has fertility 2 and produces equally often *l'Orateur* and *le président.* Later in the sentence, *starred* is connected to the phrase *marquées de un astérisque.* From an initial situation in which each French word is equally probable as a translation of *starred,* we have arrived, through training, at a situation where it is possible to connect *starred* to just the right string of four words. Near the end of the sentence, *give* is connected to *donnerai,* the first person singular future of *donner,* which means *to give.* We should be more comfortable if both *will* and *give* were connected to *donnerai,* but by limiting cepts to no more than one word, we have precluded this possibility. Finally, the last twelve words of the English sentence, *I now have the answer and will give it to the House,* clearly correspond to the last seven words of the French sentence, *je donnerai la réponse à la Chambre,* but, literally, the French is *I will give the answer to the House.* There is nothing about *now, have, and,* or *it,* and each of these words has fertility 0. Translations that are as far as this from the literal are rather more the rule than the exception in our training data. One might cavil at the connection of *la réponse* to *the answer* rather than to *it.* We do not.

1.6 Better Translation Models

Models 1 through 5 provide an effective means for obtaining word-by-word alignments of translations, but as a means to achieve our real goal, which is translation, there is room for improvement. We have seen that by ignoring the morphological structure of the two languages, we dilute the strength of our statistical model, explaining, for example, each of the several tens of forms of

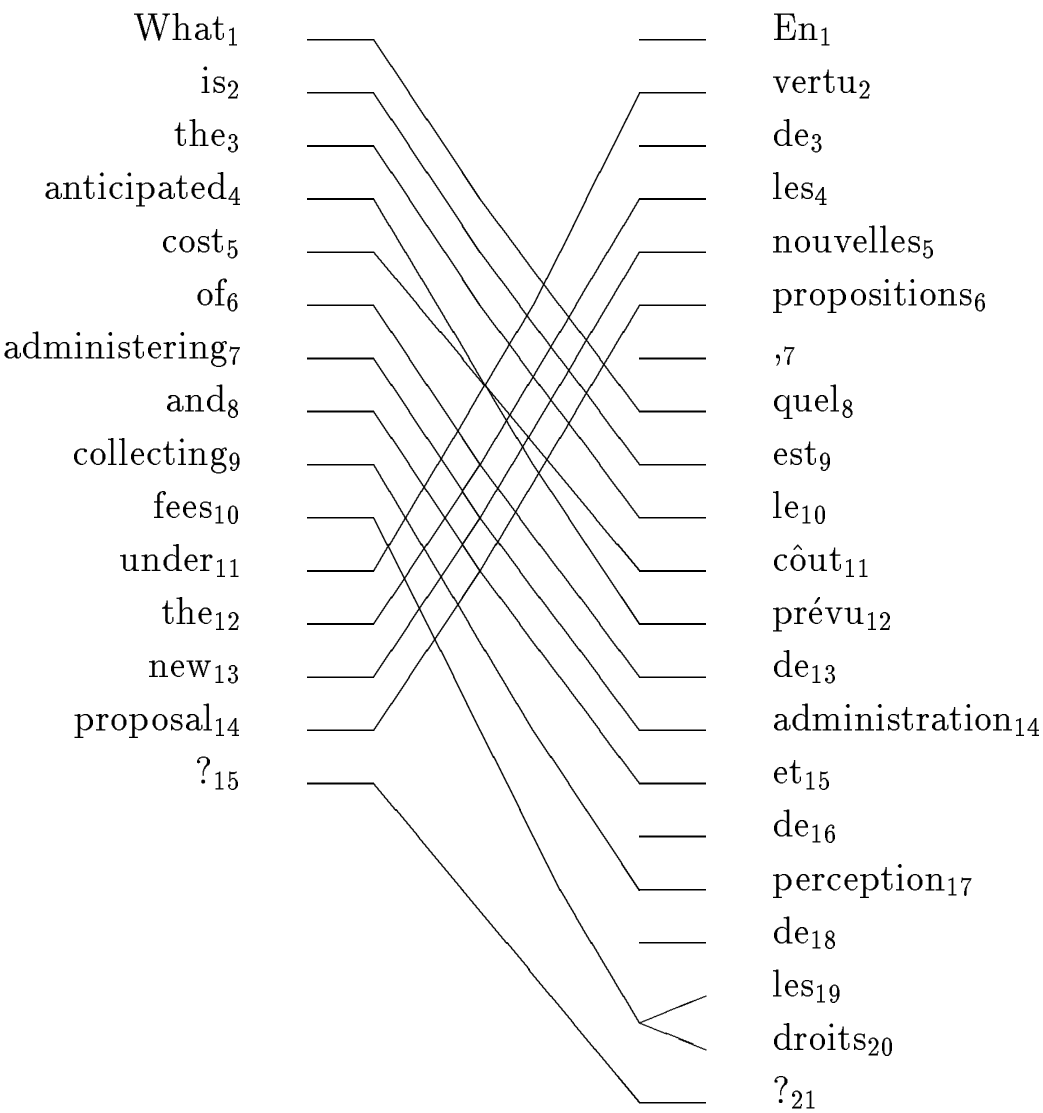

Figure 1.16: The best of 1.9×10^{25} alignments.

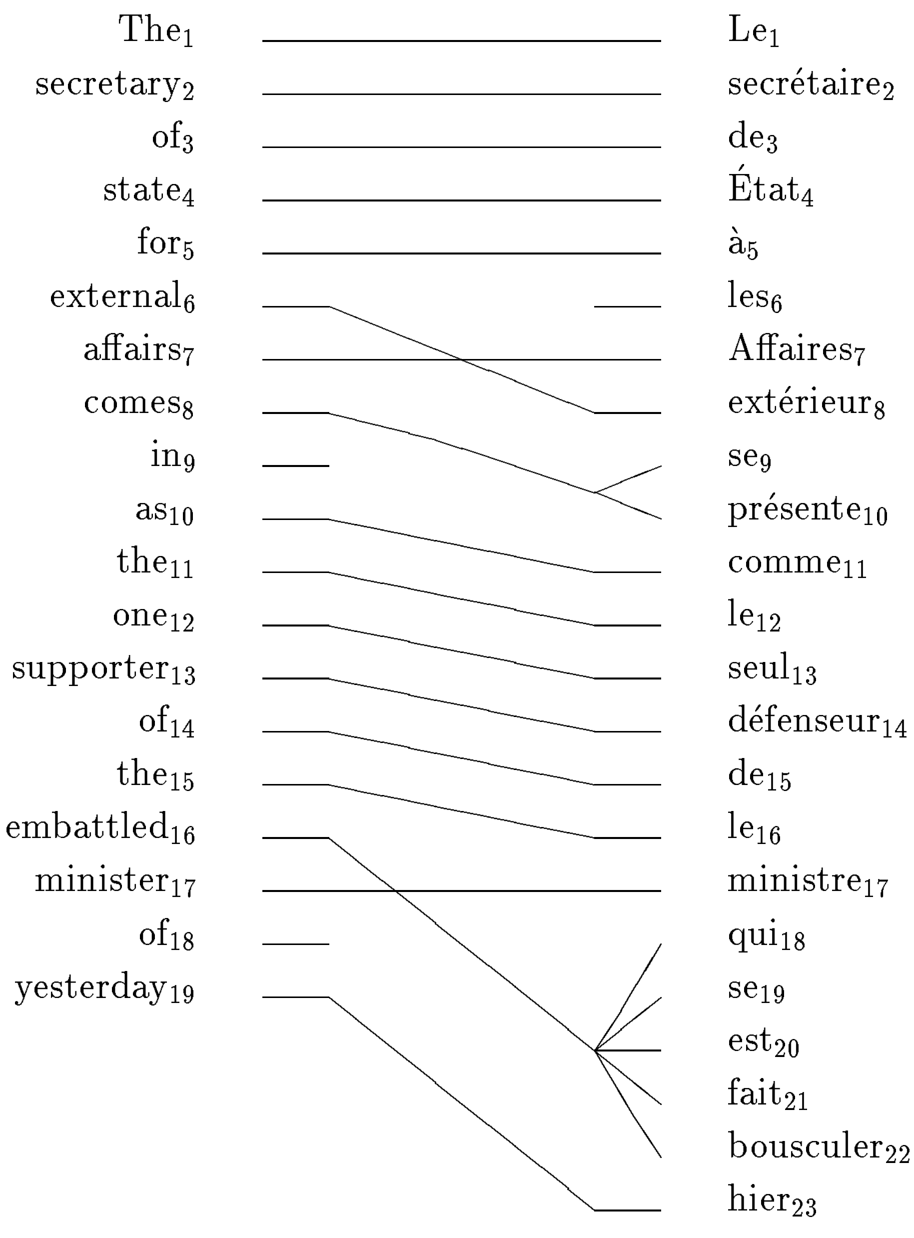

Figure 1.17: The best of 8.4×10^{29} alignments.

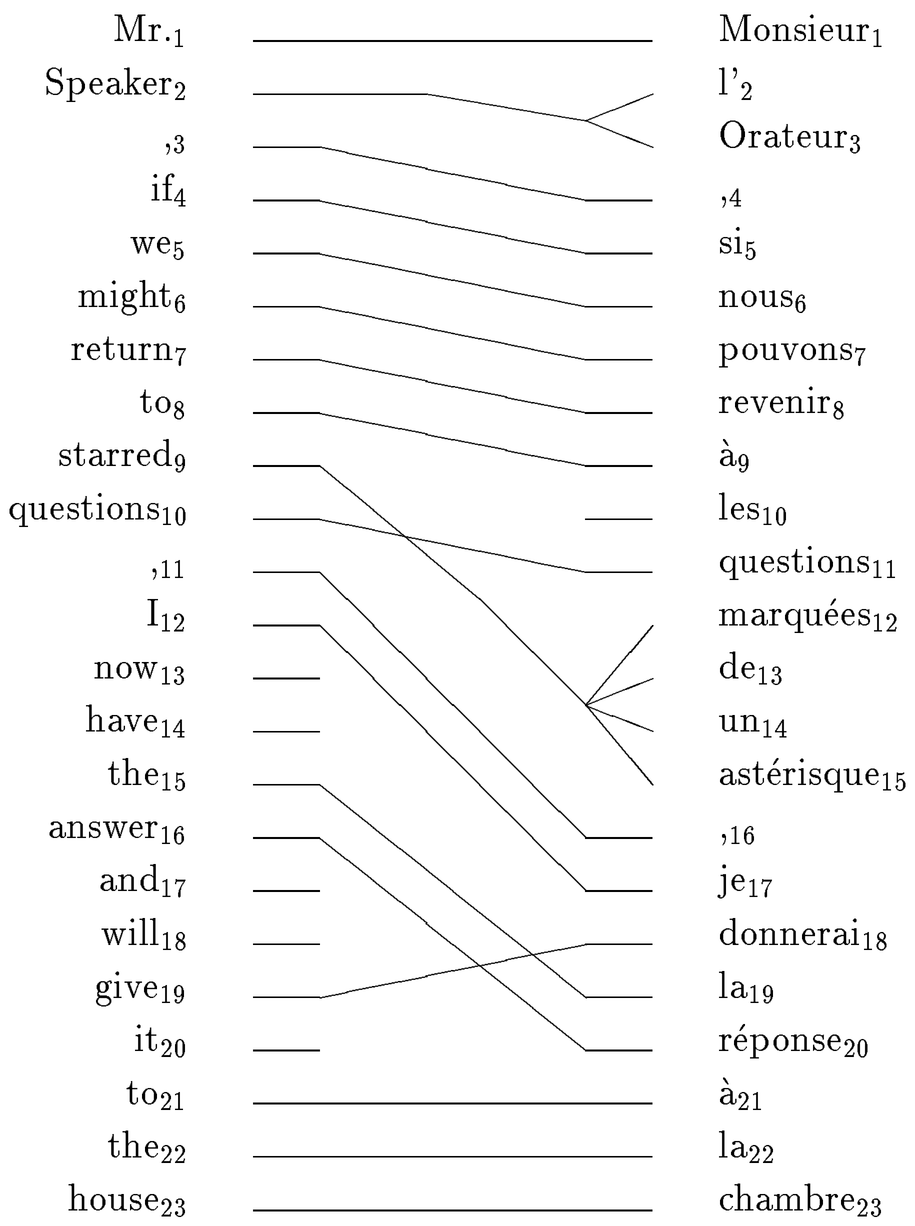

Figure 1.18: The best of 5.6×10^{31} alignments.

each French verb independently. We have seen that by ignoring multi-word cepts, we are forced to give a false, or at least an unsatisfactory, account of some features in many translations. And finally, we have seen that our models are deficient, either in fact, as with Models 3 and 4, or in spirit, as with Models 1, 2, and 5.

1.6.1 The Truth about Deficiency

We have argued in Section 1.2 that neither spiritual nor actual deficiency poses a serious problem, but this is not entirely true. Let $w(\mathbf{e})$ be the sum of $\Pr(\mathbf{f}|\mathbf{e})$ over well-formed French strings and let $i(\mathbf{e})$ be the sum over ill-formed French strings. In a deficient model, $w(\mathbf{e}) + i(\mathbf{e}) < 1$. We say that the remainder of the probability is concentrated on the event *failure* and we write $w(\mathbf{e}) + i(\mathbf{e}) + \Pr(\mathit{failure}|\mathbf{e}) = 1$. Clearly, a model is deficient precisely when $\Pr(\mathit{failure}|\mathbf{e}) > 0$. If $\Pr(\mathit{failure}|\mathbf{e}) = 0$, but $i(\mathbf{e}) > 0$, then the model is spiritually deficient. If $w(\mathbf{e})$ were independent of $\mathbf{e}$, neither form of deficiency would pose a problem, but because our models have no long-term constraints, $w(\mathbf{e})$ decreases exponentially with l. When computing alignments, even this creates no problem because $\mathbf{e}$ and $\mathbf{f}$ are known. If, however, we are given $\mathbf{f}$ and asked to discover $\hat{\mathbf{e}}$, then we will find that the product $\Pr(\mathbf{e})\Pr(\mathbf{f}|\mathbf{e})$ is too small for long English strings as compared with short ones. As a result, we will improperly favor short English strings. We can counteract this tendency in part by replacing $\Pr(\mathbf{f}|\mathbf{e})$ with $c^l \Pr(\mathbf{f}|\mathbf{e})$ for some empirically chosen constant c. This is treatment of the symptom rather than treatment of the disease itself, but it offers some temporary relief. The cure lies in better modelling.

1.6.2 Viterbi Training

As we progress from Model 1 to Model 5, evaluating the expectations that give us counts becomes increasingly difficult. For Models 1 and 2, we are able to include the contribution of each of the $(l+1)^m$ possible alignments exactly. For later models, we include the contributions of fewer and fewer alignments. Because most of the probability for each translation is concentrated by these models on a small number of alignments, this suboptimal procedure, mandated by the complexity of the models, yields acceptable results.

In the limit, we can contemplate evaluating the expectations using only a single, probable alignment for each translation. When that alignment is the Viterbi alignment, we call this *Viterbi training.* It is easy to see that Viterbi training converges: at each step, we reëstimate parameters so as to make the current set of Viterbi alignments as probable as possible; when we use these parameters to compute a new set of Viterbi alignments, we find either the old set or a set which is yet more probable. Since the probability can never be greater than one, this process must converge. In fact, unlike the EM algorithm in general, it must converge in a finite, though impractically large, number of steps because each translation has only a finite number of alignments.

In practice, we are never sure that we have found the Viterbi alignment. If we reinterpret the Viterbi alignment to mean the most probable alignment that we can find rather than the most probable alignment that exists, then a similarly reinterpreted Viterbi training algorithm still converges. We have already used this algorithm successfully as a part of a system to assign senses to English and French words on the basis of the context in which they appear [32, 31]. We expect to use it in models that we develop beyond Model 5.

1.6.3 Multi-word Cepts

In Models 1 through 5, we restrict our attention to alignments with cepts containing no more than one word each. Except in Models 4 and 5, cepts play little rôle in our development. Even in these models, cepts are determined implicitly by the fertilities of the words in the alignment: words for which the fertility is greater than zero make up one-word cepts; those for which it is zero do not. We can easily extend the generative process upon which Models 3, 4, and 5 are based to encompass multi-word cepts. We need only include a step for selecting the ceptual scheme and ascribe fertilities to cepts rather than to words, requiring that the fertility of each cept be greater than zero. Then, in Equation (1.28), we can replace the products over words in an English string with products over cepts in the ceptual scheme.

When we venture beyond one-word cepts, however, we must tread lightly. An English string can contain any of 42,005 one-word cepts, but there are more than 1.7 billion possible two-word cepts, more than 74 trillion three-word cepts, and so on. Clearly, one must be discriminating in choosing potential multi-word cepts. The caution that we have displayed thus far in limiting ourselves to cepts with fewer than two words was motivated primarily by our respect for the featureless desert that multi-word cepts offer *a priori.* The Viterbi alignments that we have computed with Model 5 give us a frame of reference from which to expand our horizons to multi-word cepts. We need only promote a multi-word sequence to cepthood when its translations as a unit, as observed in the Viterbi alignments, differ substantially from what we might expect on the basis of its component parts. In English, either a boat or a person can be left high and dry, but in French, *un bateau* is not left *haut et sec,* nor *une personne haute et sèche.* Rather, a boat is left *échoué* and a person *en plan. High and dry,* therefore, is a promising three-word cept because its translation is not compositional.

1.6.4 Morphology

We treat each distinct sequence of letters as a distinct word. In English, for example, we recognize no kinship among the several forms of the verb *to eat (eat, ate, eaten, eats,* and *eating).* In French, irregular verbs have many forms. In Figure 1.7, we have already seen seven forms of *devoir.* Altogether, it has forty-one different forms. And there would be forty-two if the French did not inexplicably drop the circumflex from the masculine plural past participle (*dus*) thereby causing it to collide with the first and second person singular in the passé sample, no doubt a source of endless confusion for the beleaguered francophone.

The French make do with fewer forms for the multitude of regular verbs that are the staple diet of everyday speech. Thus, *manger* (*to eat*), has only thirty-nine forms *(manger, mange, manges, ..., mangeassent).* Models 1 through 5 must learn to connect the five forms of *to eat* to the thirty-nine forms of *manger.* In the 28, 850, 104 French words that make up our training data, only thirteen of the thirty-nine forms of *manger* actually appear. Of course, it is only natural that in the proceedings of a parliament, forms of *manger* are less numerous than forms of *parler* (*to speak*), but even for *parler,* only twenty-eight of the thirty-nine forms occur in our data. If we were to encounter a rare form of one of these words, say, *parlassions* or *mangeassent,* we would have no inkling of its relationship to *speak* or *eat.* A similar predicament besets nouns and adjectives as well. For example, *composition* is the among the most common words in our English vocabulary, but *compositions* is among the least common words.

We plan to ameliorate these problems with a simple, inflectional analysis of verbs, nouns, adjectives, and adverbs, so that the brotherhood of the several forms of the same word is manifest in our representation of the data. For example, we wish to make evident the common pedigree of the different conjugations of a verb in French and in English; of the singular and plural, and singular possessive and plural possessive forms of a noun in English; of the singular, plural, masculine, and feminine forms of a noun or adjective in French; and of the positive, comparative, and superlative forms of an adjective or adverb in English.

Thus, our intention is to transform *(je mange la pêche | I eat the peach)* into, *e.g., (je* *manger, 13spres la pêche | I eat,x3spres the peach).* Here, *eat* is analyzed into a root, *eat,* and an ending, *x3spres,* that indicates the present tense form used except in the third person singular. Similarly, *mange* is analyzed into a root, *manger,* and an ending, *13spres,* that indicates the present tense form used for the first and third persons singular.

These transformations are invertible and should reduce the French vocabulary by about 50% and the English vocabulary by about 20%. We hope that this will significantly improve the statistics in our models.

1.7 Discussion

That interesting, bilingual, lexical correlations can be extracted automatically from a large, bilingual corpus was pointed out by Brown *et al.* [26] (Brown I in this Section). The algorithm that they describe is, roughly speaking, equivalent to carrying out the first iteration of the EM algorithm for our Model 1 starting from an initial guess in which each French word is equally probable as a translation for each English word. They were unaware of a connection to the EM algorithm, but they did realize that their method is not entirely satisfactory. For example, once it is clearly established that in *(La porte est rouge | The door is red),* it is *red* that produces *rouge,* one is uncomfortable using this sentence as support for *red* producing *porte* or *door* producing *rouge.* They suggest removing words once a correlation between them has been clearly established and then reprocessing the resulting, impoverished translations hoping to recover less obvious correlations now revealed by the departure of their more prominent relatives. From our present perspective, we see that the proper way to proceed is simply to carry out more iterations of the EM algorithm. The likelihood for Model 1 has a unique local maximum for any set of training data. As iterations proceed, the count for *porte* as a translation of *red* will dwindle away.

In a later paper, Brown *et al.* [27] (Brown II in this Section) describe a model that is essentially the same as our Model 3. They sketch the EM algorithm and show that, once trained, their model can be used to extract word-by-word alignments for pairs of sentences. They did not realize that the logarithm of the likelihood for Model 1 is concave and, hence, has a unique local maximum. They were also unaware of the trick by which we are able to sum over all alignments when evaluating the counts for Models 1 and 2, and of the trick by which we are able to sum over all alignments when transferring parameters from Model 2 to Model 3. As a result, they were unable to handle large vocabularies and so restricted themselves to vocabularies of only 9,000 words. Nonetheless, they were able to align phrases in French with the English words that produce them as illustrated in their Figure 3.

More recently, Gale and Church [51] describe an algorithm similar to the one described in Brown I. Like Brown I, they consider only the simultaneous appearance of words in pairs of sentences that are translations of one another. Although algorithms like these are extremely simple, many of the correlations between English and French words are so pronounced as to fall prey to almost any effort to expose them. Thus, the correlation of pairs like *(eau | water), (lait | milk), (pourquoi | why), (chambre | house),* and many others, simply cannot be missed. They shout from the data, and any method that is not stone deaf will hear them. But many of the correlations speak in a softer voice: to hear them clearly, we must model the translation process, as Brown I suggests and as Brown II and the current paper actually do. Only in this way can one hope to hear the quiet call of *(marquées d'un astérisque | starred)* or the whisper of *(qui s'est fait bousculer | embattled).*

The series of models that we have described constitutes a mathematical embodiment of the powerfully compelling intuitive feeling that a word in one language can be translated into a word or phrase in another language. In some cases, there may be several or even several tens of translations depending on the context in which the word appears, but we should be quite surprised to find a word with hundreds of mutually exclusive translations. Although we use these models as part of an automatic system for translating French into English, they provide, as a by-product, very satisfying accounts of the word-by-word alignment of pairs of French and English strings.

Our work has been confined to French and English, but we believe that this is purely adventitious: had the early Canadian trappers been Manchurians later to be outnumbered by swarms of conquistadores, and had the two cultures clung stubbornly each to its native tongue, we should now be aligning Spanish and Chinese. We conjecture that local alignment of the component parts of any corpus of parallel texts is inherent in the corpus itself, provided only that it be large enough. Between any pair of languages where mutual translation is important enough that the rate of accumulation of translated examples sufficiently exceeds the rate of mutation of the languages involved, there must eventually arise such a corpus.

The linguistic content of our program thus far is scant indeed. It is limited to one set of rules for analyzing a string of characters into a string of words, and another set of rules for analyzing a string of words into a string of sentences. Doubtless even these can be recast in terms of some information theoretic objective function. But it is not our intention to ignore linguistics, neither to replace it. Rather, we hope to enfold it in the embrace of a secure probabilistic framework so that the two together may draw strength from one another and guide us to better natural language processing systems in general and to better machine translation systems in particular.

Appendix A Table of Notation

$\mathcal{E}$	English vocabulary
e	English word
$\mathbf{e}$	English string
$\mathbf{E}$	random Englishstring
l	length of $\mathbf{e}$
L	random length of $\mathbf{E}$
i	position in $\mathbf{e}$, $i = 0, 1, \cdots, l$
e_i	word i of $\mathbf{e}$
e_0	the empty notion
e_1^i	$e_1 e_2 \ldots e_i$
$\mathcal{F}$	French vocabulary
f	French word
$\mathbf{f}$	French string
$\mathbf{F}$	random Frenchstring
m	length of $\mathbf{f}$
M	random length of $\mathbf{F}$
j	position in $\mathbf{f}$, $j = 1, 2, \cdots, m$
f_j	word j of $\mathbf{f}$
f_1^j	$f_1 f_2 \ldots f_j$
$\mathbf{a}$	alignment
a_j	position in $\mathbf{e}$ connected to position j of $\mathbf{f}$ for alignment $\mathbf{a}$
a_1^j	$a_1 a_2 \ldots a_j$
ϕ_i	number of positions of $\mathbf{f}$ connected to position i of $\mathbf{e}$
ϕ_1^i	$\phi_1 \phi_2 \ldots \phi_i$
τ	tableau - a sequence of tablets, where a tablet is a sequence of French words
τ_i	tablet i of τ
τ_0^i	$\tau_0 \tau_1 \ldots \tau_i$
ϕ_i	length of τ_i
k	position within a tablet, $k = 1, 2, \cdots, \phi_i$
τ_{ik}	word k of τ_i
π	a permutation of the positions of a tableau
π_{ik}	position in $\mathbf{f}$ for word k of τ_i for permutation π
${\pi_i}_1^k$	$\pi_{i1} \pi_{i2} \ldots \pi_{ik}$

$V(\mathbf{f} \mid \mathbf{e})$	Viterbi alignment for $(\mathbf{f} \mid \mathbf{e})$
$V_{i \leftarrow j}(\mathbf{f} \mid \mathbf{e})$	Viterbi alignment for $(\mathbf{f} \mid \mathbf{e})$ with ij pegged
$\mathcal{N}(\mathbf{a})$	neighboring alignments of $\mathbf{a}$
$\mathcal{N}_{ij}(\mathbf{a})$	neighboring alignments of $\mathbf{a}$ with ij pegged
$b(\mathbf{a})$	alignment in $\mathcal{N}(\mathbf{a})$ with greatest probability
$b^{\infty}(\mathbf{a})$	alignment obtained by applying b repeatedly to $\mathbf{a}$
$b_{i \leftarrow j}(\mathbf{a})$	alignment in $\mathcal{N}_{ij}(\mathbf{a})$ with greatest probability
$b^{\infty}_{i \leftarrow j}(\mathbf{a})$	alignment obtained by applying $b_{i \leftarrow j}$ repeatedly to $\mathbf{a}$
$\mathcal{A}(e)$	class of English word e
$\mathcal{B}(f)$	class of French word f
Δj	displacement of a word in $\mathbf{f}$
v	vacancies in $\mathbf{f}$
ρ_i	first position in $\mathbf{e}$ to the left of i that has non-zero fertility
c_i	average position in $\mathbf{f}$ of the words connected to position i of $\mathbf{e}$
$[i]$	position in $\mathbf{e}$ of the i^{th} one word notion
$\odot_i$	$c_{[i]}$
P_θ	translation model P with parameter values θ
$\mathrm{C}(\mathbf{f}, \mathbf{e})$	empirical distribution of a sample
$\psi(\mathrm{P}_\theta)$	log-likelihood objective function
$R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta)$	relative objective function
$t(f \mid e)$	translation probabilities (All Models)
$\epsilon(m \mid l)$	string length probabilities (Models 1 and 2)
$n(\phi \mid e)$	fertility probabilities (Models 3, 4, and 5)
p_0, p_1	fertility probabilities for e_0 (Models 3, 4, and 5)
$a(i \mid j, l, m)$	alignment probabilities (Model 2)
$d(j \mid i, l, m)$	distortion probabilities (Model 3)
$d_1(\Delta j \mid \mathcal{A}, \mathcal{B})$	distortion probabilities for the first word of a tablet (Model 4)
$d_{>1}(\Delta j \mid \mathcal{B})$	distortion probabilities for the other words of a tablet (Model 4)
$d_1(\Delta j \mid \mathcal{B}, v)$	distortion probabilities for the first word of a tablet (Model 5)
$d_{>1}(\Delta j \mid \mathcal{B}, v)$	distortion probabilities for the other words of a tablet (Model 5)

Appendix B Summary of Models

We collect here brief descriptions of our various translation models and the formulae needed for training them.

1.2.1 Translation Models

An *English-to-French translation model* P with parameters θ is a formula for calculating a conditional probability, or likelihood, $\mathrm{P}_\theta(\mathbf{f} \mid \mathbf{e})$, for any string $\mathbf{f}$ of French words and any string $\mathbf{e}$ of English words. These probabilities satisfy

$$\begin{aligned} &\mathrm{P}_\theta(\mathbf{f} \mid \mathbf{e}) \geq 0, \qquad \mathrm{P}_\theta(\textit{failure} \mid \mathbf{e}) \geq 0, \\ &\mathrm{P}_\theta(\textit{failure} \mid \mathbf{e}) + \sum_{\mathbf{f}} \mathrm{P}_\theta(\mathbf{f} \mid \mathbf{e}) = 1, \end{aligned} \tag{1.49}$$

where the sum ranges over all French strings $\mathbf{f}$, and *failure* is a special symbol not in the French vocabulary. We interpret $\mathrm{P}_\theta(\mathbf{f} \mid \mathbf{e})$ as the probability that a translator will produce $\mathbf{f}$ when given $\mathbf{e}$, and $\mathrm{P}_\theta(\textit{failure} \mid \mathbf{e})$ as the probability that he will produce no translation when given $\mathbf{e}$. We call a model *deficient* if $\mathrm{P}_\theta(\textit{failure} \mid \mathbf{e})$ is greater than zero for some $\mathbf{e}$.

Log-Likelihood Objective Function

The *log-likelihood* of a sample of translations $(\mathbf{f}^{(s)}, \mathbf{e}^{(s)})$, $s = 1, 2, \cdots, S$, is

$$\psi(\mathrm{P}_\theta) \quad = \quad S^{-1} \sum_{s=1}^{S} \log \mathrm{P}_\theta(\mathbf{f}^{(s)} \mid \mathbf{e}^{(s)}) = \sum_{\mathbf{f},\mathbf{e}} \mathrm{C}(\mathbf{f}, \mathbf{e}) \log \mathrm{P}_\theta(\mathbf{f} \mid \mathbf{e}). \tag{1.50}$$

Here $\mathrm{C}(\mathbf{f}, \mathbf{e})$ is the empirical distribution of the sample, so that $\mathrm{C}(\mathbf{f}, \mathbf{e})$ is $1/S$ times the number of times (usually 0 or 1) that the pair $(\mathbf{f}, \mathbf{e})$ occurs in the sample. We determine values for the parameters θ so as to maximize this log-likelihood for a large *training sample* of translations.

Hidden Alignment Models

For the models that we present here, we can express $\mathrm{P}_\theta(\mathbf{f} \mid \mathbf{e})$ as a sum of the probabilities of *hidden alignments* $\mathbf{a}$ between $\mathbf{e}$ and $\mathbf{f}$:

$$\mathrm{P}_\theta(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} \mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e}). \tag{1.51}$$

For our models, the only alignments that have positive probability are those for which each word of $\mathbf{f}$ is connected to at most one word of $\mathbf{e}$.

Relative Objective Function

We can compare hidden alignment models $\tilde{\mathrm{P}}_{\tilde{\theta}}$ and P_θ using the *relative objective function* *

$$R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta) \quad \equiv \quad \sum_{\mathbf{f},\mathbf{e}} \mathrm{C}(\mathbf{f}, \mathbf{e}) \sum_{\mathbf{a}} \tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{a} \mid \mathbf{f}, \mathbf{e}) \log \frac{\mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e})}{\tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{f}, \mathbf{a} \mid \mathbf{e})}, \tag{1.53}$$

where $\tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{a} \mid \mathbf{f}, \mathbf{e}) = \tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{a}, \mathbf{f} \mid \mathbf{e}) / \tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{f} \mid \mathbf{e})$. Note that $R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \tilde{\mathrm{P}}_{\tilde{\theta}}) = 0$. R is related to ψ by Jensen's inequality

$$\psi(\mathrm{P}_\theta) \geq \psi(\tilde{\mathrm{P}}_{\tilde{\theta}}) + R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta), \tag{1.54}$$

*The reader will notice a similarity between $R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta)$ and the *relative entropy*

$$D(p, q) = \sum_{x} p(x) \log \frac{p(x)}{q(x)} \tag{1.52}$$

between probability distributions p and q. However, whereas the relative entropy is never negative, R can take any value. The inequality (1.54) for R is the analog of the inequality $D \geq 0$ for D.

which follows because the logarithm is concave. In fact, for any $\mathbf{e}$ and $\mathbf{f}$,

$$\sum_{\mathbf{a}} \tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{a} \mid \mathbf{f}, \mathbf{e}) \log \frac{\mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e})}{\tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{f}, \mathbf{a} \mid \mathbf{e})} \leq \log \sum_{\mathbf{a}} \tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{a} \mid \mathbf{f}, \mathbf{e}) \frac{\mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e})}{\tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{f}, \mathbf{a} \mid \mathbf{e})} \tag{1.55}$$

$$= \log \frac{\mathrm{P}_\theta(\mathbf{f} \mid \mathbf{e})}{\tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{f} \mid \mathbf{e})} = \log \mathrm{P}_\theta(\mathbf{f} \mid \mathbf{e}) - \log \tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{f} \mid \mathbf{e}). \tag{1.56}$$

Summing over $\mathbf{e}$ and $\mathbf{f}$ and using the Definitions (1.50) and (1.53) we arrive at Equation (1.54).

1.2.2 Iterative Improvement

We cannot create a good model or find good parameter values at a stroke. Rather we employ a process of iterative improvement. For a given model we use current parameter values to find better ones, and in this way, from initial values we find locally optimal ones. Then, given good parameter values for one model, we use them to find initial parameter values for another model. By alternating between these two steps we proceed through a sequence of gradually more sophisticated models.

Improving Parameter Values

From Jensen's inequality (1.54), we see that $\psi(\mathrm{P}_\theta)$ is greater than $\psi(\tilde{\mathrm{P}}_{\tilde{\theta}})$ if $R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta)$ is positive. With $\tilde{\mathrm{P}} = \mathrm{P}$, this suggests the following iterative procedure, known as the *EM Algorithm* [15, 46], for finding locally optimal parameter values θ for a model P:

0. Choose some initial values $\tilde{\theta}$.

1. Repeat Steps $2 - 3$ until convergence.

2. With $\tilde{\theta}$ fixed, find the values θ which maximize $R(\mathrm{P}_{\tilde{\theta}}, \mathrm{P}_\theta)$.

3. Replace $\tilde{\theta}$ by θ.

Note that for any $\tilde{\theta}$, $R(\mathrm{P}_{\tilde{\theta}}, \mathrm{P}_\theta)$ is non-negative at its maximum in θ, since it is zero for $\theta = \tilde{\theta}$. Thus $\psi(\mathrm{P}_\theta)$ will not decrease from one iteration to the next.

Going From One Model to Another

Jensen's inequality also suggests a method for using parameter values $\tilde{\theta}$ for one model $\tilde{\mathrm{P}}$ to find initial parameter values θ for another model P:

4. With $\tilde{\mathrm{P}}$ and $\tilde{\theta}$ fixed, find the values θ which maximize $R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta)$.

In contrast to the case where $\tilde{\mathrm{P}} = \mathrm{P}$, there may not be any θ for which $R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta)$ is non-negative. Thus, it could be that, even for the best θ, $\psi(\mathrm{P}_\theta) < \psi(\tilde{\mathrm{P}}_{\tilde{\theta}})$.

Parameter Reestimation Formulae

In order to apply these algorithms, we need to solve the maximization problems of Steps 2 and 4. For the models that we consider, we can do this explicitly. To exhibit the basic form of the solution, we suppose P_θ is a model given by

$$\mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = \prod_{\omega \in \Omega} \theta(\omega)^{c(\omega\,;\,\mathbf{a}, \mathbf{f}, \mathbf{e})}, \tag{1.57}$$

where the $\theta(\omega), \omega \in \Omega$, are real-valued parameters satisfying the constraints

$$\theta(\omega) \geq 0, \qquad\qquad \sum_{\omega \in \Omega} \theta(\omega) = 1, \tag{1.58}$$

and for each ω and $(\mathbf{a}, \mathbf{f}, \mathbf{e})$, $c(\omega\,; \mathbf{a}, \mathbf{f}, \mathbf{e})$ is a non-negative integer.[†] We interpret $\theta(\omega)$ as the probability of the event ω and $c(\omega\,; \mathbf{a}, \mathbf{f}, \mathbf{e})$ as the number of times that this event occurs in $(\mathbf{a}, \mathbf{f}, \mathbf{e})$. Note that

$$c(\omega\,; \mathbf{a}, \mathbf{f}, \mathbf{e}) = \theta(\omega) \frac{\partial}{\partial \theta(\omega)} \log \mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e}). \tag{1.59}$$

The values for θ that maximize the relative objective function $R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta)$ subject to the contraints (1.58) are determined by the Kuhn-Tucker conditions

$$\frac{\partial}{\partial \theta(\omega)} R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta) - \lambda = 0, \qquad \omega \in \Omega, \tag{1.60}$$

where λ is a Lagrange multiplier, the value of which is determined by the equality constraint in Equation (1.58). These conditions are both necessary and sufficient for a maximum since $R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta)$ is a concave function of the $\theta(\omega)$. By multiplying Equation (1.60) by $\theta(\omega)$ and using Equation (1.59) and Definition (1.53) of R, we obtain the parameter reestimation formulae

$$\begin{aligned}
\theta(\omega) &= \lambda^{-1} \tilde{c}_{\tilde{\theta}}(\omega), \qquad\qquad \lambda = \sum_{\omega \in \Omega} \tilde{c}_{\tilde{\theta}}(\omega), && (1.61)\\
\tilde{c}_{\tilde{\theta}}(\omega) &= \sum_{\mathbf{f}, \mathbf{e}} \mathrm{C}(\mathbf{f}, \mathbf{e}) \tilde{c}_{\tilde{\theta}}(\omega\,; \mathbf{f}, \mathbf{e}), && (1.62)\\
\tilde{c}_{\tilde{\theta}}(\omega\,; \mathbf{f}, \mathbf{e}) &= \sum_{\mathbf{a}} \tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{a} \mid \mathbf{f}, \mathbf{e}) c(\omega\,; \mathbf{a}, \mathbf{f}, \mathbf{e}). && (1.63)
\end{aligned}$$

We interpret $\tilde{c}_{\tilde{\theta}}(\omega\,; \mathbf{f}, \mathbf{e})$ as the expected number of times, as computed by the model $\tilde{\mathrm{P}}_{\tilde{\theta}}$, that the event ω occurs in the translation of $\mathbf{e}$ to $\mathbf{f}$. Thus $\theta(\omega)$ is the (normalized) expected number of times, as computed by model $\tilde{\mathrm{P}}_{\tilde{\theta}}$, that ω occurs in the translations of the training sample.

We can easily generalize these formulae to allow models of the form (1.57) for which the single equality constraint in Equation (1.58) is replaced by multiple constraints

$$\sum_{\omega \in \Omega_\mu} \theta(\omega) = 1, \qquad \mu = 1, 2, \cdots, \tag{1.64}$$

where the subsets Ω_μ, $\mu = 1, 2, \cdots$ form a partition of Ω. We need only modify Equation (1.61) by allowing a different λ_μ for each μ: if $\omega \in \Omega_\mu$, then

$$\theta(\omega) = {\lambda_\mu}^{-1} \tilde{c}_{\tilde{\theta}}(\omega), \qquad\qquad \lambda_\mu = \sum_{\omega \in \Omega_\mu} \tilde{c}_{\tilde{\theta}}(\omega). \tag{1.65}$$

1.2.3 Model 1

Parameters

$\epsilon(m \mid l)$	string length probabilities
$t(f \mid e)$	translation probabilities

Here $f \in \mathcal{F}$; $e \in \mathcal{E}$ or $e = e_0$; $l = 1, 2, \cdots$; and $m = 1, 2, \cdots$.

General Formula

$$\mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = \mathrm{P}_\theta(m \mid \mathbf{e}) \mathrm{P}_\theta(\mathbf{a} \mid m, \mathbf{e}) \mathrm{P}_\theta(\mathbf{f} \mid \mathbf{a}, m, \mathbf{e}) \tag{1.66}$$

[†]More generally, we can allow $c(\omega\,; \mathbf{a}, \mathbf{f}, \mathbf{e})$ to be a non-negative real number.

Assumptions

$$\begin{aligned}
\mathrm{P}_\theta(m \mid \mathbf{e}) &= \epsilon(m \mid l) && (1.67)\\
\mathrm{P}_\theta(\mathbf{a} \mid m, \mathbf{e}) &= (l+1)^{-m} && (1.68)\\
\mathrm{P}_\theta(\mathbf{f} \mid \mathbf{a}, m, \mathbf{e}) &= \prod_{j=1}^{m} t(f_j \mid e_{a_j}) && (1.69)
\end{aligned}$$

This model is not deficient.

Generation

Equations (1.66) - (1.69) describe the following process for producing $\mathbf{f}$ from $\mathbf{e}$:

1. Choose a length m for $\mathbf{f}$ according to the probability distribution $\epsilon(m \mid l)$.
2. For each $j = 1, 2, \cdots, m$, choose a_j from $0, 1, 2, \cdots l$ according to the uniform distribution.
3. For each $j = 1, 2, \cdots, m$, choose a French word f_j according to the distribution $t(f_j \mid e_{a_j})$.

Useful Formulae

Because of the independence assumptions (1.67)-(1.69), we can calculate the sum over alignments (1.51) in closed form:

$$\begin{aligned}
\mathrm{P}_\theta(\mathbf{f} \mid \mathbf{e}) &= \sum_{\mathbf{a}} \mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) && (1.70)\\
&= \epsilon(m \mid l)(l+1)^{-m} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j \mid e_{a_j}) && (1.71)\\
&= \epsilon(m \mid l)(l+1)^{-m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j \mid e_i). && (1.72)
\end{aligned}$$

Equation (1.72) is useful in computations since it involves only $O(lm)$ arithmetic operations, whereas the original sum over alignments (1.71) involves $O(l^m)$ operations.

Concavity

The objective function (1.50) for this model is a strictly concave function of the parameters. In fact, from Equations (1.50) and (1.72),

$$\psi(\mathrm{P}_\theta) = \sum_{\mathbf{f},\mathbf{e}} \mathrm{C}(\mathbf{f}, \mathbf{e}) \sum_{j=1}^{m} \log \sum_{i=0}^{l} t(f_j \mid e_i) + \sum_{\mathbf{f},\mathbf{e}} \mathrm{C}(\mathbf{f}, \mathbf{e}) \log \epsilon(m \mid l) + \mathit{constant} \qquad (1.73)$$

which is clearly concave in $\epsilon(m \mid l)$ and $t(f \mid e)$ since the logarithm of a sum is concave, and the sum of concave functions is concave.

Because ψ is concave, it has a unique *local maximum*. Moreover, we will find this maximum using the EM algorithm, provided that none of our initial parameter values is zero.

1.2.4 Model 2

Parameters

$\epsilon(m \mid l)$	string length probabilities
$t(f \mid e)$	translation probabilities
$a(i \mid j, l, m)$	alignment probabilities

Here $i = 0, \cdots, l$; and $j = 1, \cdots, m$.

General Formula

$$\mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = \mathrm{P}_\theta(m \mid \mathbf{e})\mathrm{P}_\theta(\mathbf{a} \mid m, \mathbf{e})\mathrm{P}_\theta(\mathbf{f} \mid \mathbf{a}, m, \mathbf{e}) \tag{1.74}$$

Assumptions

$$\mathrm{P}_\theta(m \mid \mathbf{e}) = \epsilon(m \mid l) \tag{1.75}$$

$$\mathrm{P}_\theta(\mathbf{a} \mid m, \mathbf{e}) = \prod_{j=1}^{m} a(a_j \mid j, l, m) \tag{1.76}$$

$$\mathrm{P}_\theta(\mathbf{f} \mid \mathbf{a}, m, \mathbf{e}) = \prod_{j=1}^{m} t(f_j \mid e_{a_j}) \tag{1.77}$$

This model is not deficient. Model 1 is the special case of this model in which the alignment probabilities are uniform: $a(i \mid j, l, m) = (l+1)^{-1}$ for all i.

Generation

Equations (1.74) - (1.77) describe the following process for producing $\mathbf{f}$ from $\mathbf{e}$:

1. Choose a length m for $\mathbf{f}$ according to the distribution $\epsilon(m \mid l)$.
2. For each $j = 1, 2, \cdots, m$, choose a_j from $0, 1, 2, \cdots l$ according to the distribution $a(a_j \mid j, l, m)$.
3. For each j, choose a French word f_j according to the distribution $t(f_j \mid e_{a_j})$.

Useful Formulae

Just as for Model 1, the independence assumptions allow us to calculate the sum over alignments (1.51) in closed form:

$$\mathrm{P}_\theta(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a}} \mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) \tag{1.78}$$

$$= \epsilon(m \mid l) \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} t(f_j \mid e_{a_j}) a(a_j \mid j, l, m) \tag{1.79}$$

$$= \epsilon(m \mid l) \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j \mid e_i) a(i \mid j, l, m). \tag{1.80}$$

By assumption (1.76) the connections of $\mathbf{a}$ are independent given the length m of $\mathbf{f}$. Using Equation (1.80) we find that they are also independent given $\mathbf{f}$:

$$\mathrm{P}_\theta(\mathbf{a} \mid \mathbf{f}, \mathbf{e}) = \prod_{j=1}^{m} p_\theta(a_j \mid j, \mathbf{f}, \mathbf{e}), \tag{1.81}$$

where

$$p_\theta(i \mid j, \mathbf{f}, \mathbf{e}) = \frac{\gamma(i, j, \mathbf{f}, \mathbf{e})}{\sum_{i'} \gamma(i', j, \mathbf{f}, \mathbf{e})} \qquad \text{with} \qquad \gamma(i, j, \mathbf{f}, \mathbf{e}) = t(f_j \mid e_i) a(i \mid j, l, m). \tag{1.82}$$

Viterbi Alignment

For this model, and thus also for Model 1, we can express in closed form the Viterbi alignment $V(\mathbf{f} \mid \mathbf{e})$ between a pair of strings $(\mathbf{f}, \mathbf{e})$:

$$V(\mathbf{f} \mid \mathbf{e})_j = \operatorname*{argmax}_i t(f_j \mid e_i) a(i \mid j, l, m). \tag{1.83}$$

Parameter Reestimation Formulae

We can find the parameter values θ that maximize the relative objective function $R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta)$ by applying the considerations of Section 1.2.2. The counts $c(\omega\,; \mathbf{a}, \mathbf{f}, \mathbf{e})$ of Equation (1.57) are

$$\begin{aligned} c(f \mid e\,; \mathbf{a}, \mathbf{f}, \mathbf{e}) &= \sum_{j=1}^{m} \delta(e, e_{a_j}) \delta(f, f_j), & (1.84) \\ c(i \mid j, l, m\,; \mathbf{a}, \mathbf{f}, \mathbf{e}) &= \delta(i, a_j). & (1.85) \end{aligned}$$

We obtain the parameter reestimation formulae for $t(f \mid e)$ and $a(i \mid j, l, m)$ by using these counts in Equations (1.61) - (1.65).

Equation (1.63) requires a sum over alignments. If $\tilde{\mathrm{P}}_{\tilde{\theta}}$ satisfies

$$\tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{a} \mid \mathbf{f}, \mathbf{e}) = \prod_{j=1}^{m} \tilde{p}_{\tilde{\theta}}(a_j \mid j, \mathbf{f}, \mathbf{e}), \tag{1.86}$$

as is the case for Models 1 and 2 (see Equation (1.81)), then this sum can be calculated explicitly:

$$\begin{aligned} \tilde{c}_{\tilde{\theta}}(f \mid e\,; \mathbf{f}, \mathbf{e}) &= \sum_{i=0}^{l} \sum_{j=1}^{m} \tilde{p}_{\tilde{\theta}}(i \mid j, \mathbf{f}, \mathbf{e}) \delta(e, e_i) \delta(f, f_j), & (1.87) \\ \tilde{c}_{\tilde{\theta}}(i \mid j\,; \mathbf{f}, \mathbf{e}) &= \tilde{p}_{\tilde{\theta}}(i \mid j, \mathbf{f}, \mathbf{e}). & (1.88) \end{aligned}$$

Equations (1.84) - (1.88) involve only $O(lm)$ arithmetic operations, whereas the sum over alignments involves $O(l^m)$ operations.

1.2.5 Model 3

Parameters

$t(f \mid e)$	translation probabilities
$n(\phi \mid e)$	fertility probabilities
p_0, p_1	fertility probabilities for e_0
$d(j \mid i, l, m)$	distortion probabilities

Here $\phi = 0, 1, 2, \cdots$.

General Formulae

$$\begin{aligned} \mathrm{P}_\theta(\tau, \pi \mid \mathbf{e}) &= \mathrm{P}_\theta(\phi \mid \mathbf{e}) \mathrm{P}_\theta(\tau \mid \phi, \mathbf{e}) \mathrm{P}_\theta(\pi \mid \tau, \phi, \mathbf{e}) & (1.89) \\ \mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) &= \sum_{(\tau,\pi) \in \langle \mathbf{f}, \mathbf{a} \rangle} \mathrm{P}_\theta(\tau, \pi \mid \mathbf{e}) & (1.90) \end{aligned}$$

Here $\langle \mathbf{f}, \mathbf{a} \rangle$ is the set of all (τ, π) *consistent* with $(\mathbf{f}, \mathbf{a})$:

$$\begin{aligned} &(\tau, \pi) \in \langle \mathbf{f}, \mathbf{a} \rangle \text{ if for all } i = 0, 1, \cdots, l \text{ and } k = 1, 2, \cdots, \phi_i, \\ &f_{\pi_{ik}} = \tau_{ik} \text{ and } a_{\pi_{ik}} = i. \end{aligned} \tag{1.91}$$

Assumptions

$$\begin{aligned} \mathrm{P}_\theta(\phi \mid \mathbf{e}) &= n_0(\phi_0 \mid \sum_{i=1}^{l} \phi_i) \prod_{i=1}^{l} n(\phi_i \mid e_i) & (1.92) \\ \mathrm{P}_\theta(\tau \mid \phi, \mathbf{e}) &= \prod_{i=0}^{l} \prod_{k=1}^{\phi_i} t(\tau_{ik} \mid e_i) & (1.93) \\ \mathrm{P}_\theta(\pi \mid \tau, \phi, \mathbf{e}) &= \frac{1}{\phi_0!} \prod_{i=1}^{l} \prod_{k=1}^{\phi_i} d(\pi_{ik} \mid i, l, m) & (1.94) \end{aligned}$$

where

$$n_0(\phi_0 \mid m') = \binom{m'}{\phi_0} p_0{}^{m'-\phi_0} p_1{}^{\phi_0}. \tag{1.95}$$

In Equation (1.94) the factor of $1/\phi_0!$ accounts for the choices of π_{0k}, $k = 1, 2, \cdots, \phi_0$. This model is deficient, since

$$\mathrm{P}_\theta(\mathit{failure} \mid \tau, \phi, \mathbf{e}) \equiv 1 - \sum_\pi \mathrm{P}_\theta(\pi \mid \tau, \phi, \mathbf{e}) > 0. \tag{1.96}$$

Generation

Equations (1.89) - (1.94) describe the following process for producing $\mathbf{f}$ or *failure* from $\mathbf{e}$:

1. For each $i = 1, 2, \cdots, l$, choose a length ϕ_i for τ_i according to the distribution $n(\phi_i \mid e_i)$.
2. Choose a length ϕ_0 for τ_0 according to the distribution $n_0(\phi_0 \mid \sum_{i=1}^l \phi_i)$.
3. Let $m = \phi_0 + \sum_{i=1}^l \phi_i$.
4. For each $i = 0, 1, \cdots, l$ and each $k = 1, 2, \cdots, \phi_i$, choose a French word τ_{ik} according to the distribution $t(\tau_{ik} \mid e_i)$.
5. For each $i = 1, 2, \cdots, l$ and each $k = 1, 2, \cdots, \phi_i$, choose a position π_{ik} from $1, \cdots, m$ according to the distribution $d(\pi_{ik} \mid i, l, m)$.
6. If any position has been chosen more than once then return *failure*.

1. For each $k = 1, 2, \cdots, \phi_0$, choose a position π_{0k} from the $\phi_0 - k + 1$ remaining vacant positions in $1, 2, \cdots, m$ according to the uniform distribution.

8. Let $\mathbf{f}$ be the string with $f_{\pi_{ik}} = \tau_{ik}$.

Useful Formulae

From Equations (1.92) - (1.94) it follows that if (τ, π) are consistent with $(\mathbf{f}, \mathbf{a})$ then

$$\mathrm{P}_\theta(\tau \mid \phi, \mathbf{e}) = \prod_{j=1}^m t(f_j \mid e_{a_j}), \tag{1.97}$$

$$\mathrm{P}_\theta(\pi \mid \tau, \phi, \mathbf{e}) = \frac{1}{\phi_0!} \prod_{j:a_j \neq 0} d(j \mid a_j, l, m). \tag{1.98}$$

In Equation (1.98), the product runs over all $j = 1, 2, \cdots, m$ except those for which $a_j = 0$.

By summing over all pairs (τ, π) consistent with $(\mathbf{f}, \mathbf{a})$ we find

$$\mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = \sum_{(\tau,\pi) \in \langle \mathbf{f}, \mathbf{a} \rangle} \mathrm{P}_\theta(\tau, \pi \mid \mathbf{e}) \tag{1.99}$$

$$= n_0(\phi_0 \mid \sum_{i=1}^l \phi_i) \prod_{i=1}^l n(\phi_i \mid e_i) \phi_i! \prod_{j=1}^m t(f_j \mid e_{a_j}) \prod_{j:a_j \neq 0} d(j \mid a_j, l, m). \tag{1.100}$$

The factors of $\phi_i!$ in Equation (1.100) arise because there are $\prod_{i=0}^l \phi_i!$ equally probable terms in the sum (1.99).

Parameter Reestimation Formulae

We can find the parameter values θ that maximize the relative objective function $R(\tilde{\mathrm{P}}_{\tilde{\theta}}, \mathrm{P}_\theta)$ by applying the considerations of Section 1.2.2. The counts $c(\omega\,; \mathbf{a}, \mathbf{f}, \mathbf{e})$ of Equation (1.57) are

$$c(f \mid e\,; \mathbf{a}, \mathbf{f}, \mathbf{e}) = \sum_{j=1}^{m} \delta(e, e_{a_j})\delta(f, f_j), \tag{1.101}$$

$$c(j \mid i, l, m\,; \mathbf{a}, \mathbf{f}, \mathbf{e}) = \delta(i, a_j), \tag{1.102}$$

$$c(\phi \mid e\,; \mathbf{a}, \mathbf{f}, \mathbf{e}) = \sum_{i=1}^{l} \delta(e, e_i)\delta(\phi, \phi_i). \tag{1.103}$$

We obtain the parameter reestimation formulae for $t(f \mid e)$, $a(j \mid i, l, m)$, and $t(\phi \mid e)$ by using these counts in Equations (1.61) - (1.65).

Equation (1.63) requires a sum over alignments. If $\tilde{\mathrm{P}}_{\tilde{\theta}}$ satisfies

$$\tilde{\mathrm{P}}_{\tilde{\theta}}(\mathbf{a} \mid \mathbf{f}, \mathbf{e}) = \prod_{j=1}^{m} \tilde{p}_{\tilde{\theta}}(a_j \mid j, \mathbf{f}, \mathbf{e}), \tag{1.104}$$

as is the case for Models 1 and 2 (see Equation (1.81)), then this sum can be calculated explicitly for $\tilde{c}_{\tilde{\theta}}(f \mid e\,; \mathbf{f}, \mathbf{e})$ and $\tilde{c}_{\tilde{\theta}}(j \mid i\,; \mathbf{f}, \mathbf{e})$:

$$\tilde{c}_{\tilde{\theta}}(f \mid e\,; \mathbf{f}, \mathbf{e}) = \sum_{i=0}^{l}\sum_{j=1}^{m} \tilde{p}_{\tilde{\theta}}(i \mid j, \mathbf{f}, \mathbf{e})\delta(e, e_i)\delta(f, f_j), \tag{1.105}$$

$$\tilde{c}_{\tilde{\theta}}(j \mid i\,; \mathbf{f}, \mathbf{e}) = \tilde{p}_{\tilde{\theta}}(i \mid j, \mathbf{f}, \mathbf{e}). \tag{1.106}$$

Unfortunately, there is no analogous formula for $\tilde{c}_{\tilde{\theta}}(\phi \mid e\,; \mathbf{f}, \mathbf{e})$. Instead we must be content with

$$\tilde{c}_{\tilde{\theta}}(\phi \mid e\,; \mathbf{f}, \mathbf{e}) = \sum_{i=1}^{l} \delta(e, e_i) \prod_{j=1}^{m}(1 - \tilde{p}_{\tilde{\theta}}(i \mid j, \mathbf{f}, \mathbf{e})) \sum_{\gamma \in \Gamma_\phi} \prod_{k=1}^{\phi} \frac{\alpha_{ik}(\mathbf{f}, \mathbf{e})^{\gamma_k}}{\gamma_k!}, \tag{1.107}$$

$$\alpha_{ik}(\mathbf{f}, \mathbf{e}) = \frac{(-1)^{k+1}}{k} \sum_{j=1}^{m} \beta_{ij}(\mathbf{f}, \mathbf{e})^k, \tag{1.108}$$

$$\beta_{ij}(\mathbf{f}, \mathbf{e}) = \frac{\tilde{p}_{\tilde{\theta}}(i \mid j, \mathbf{f}, \mathbf{e})}{1 - \tilde{p}_{\tilde{\theta}}(i \mid j, \mathbf{f}, \mathbf{e})}. \tag{1.109}$$

In Equation (1.107), Γ_ϕ denotes the set of all partitions of ϕ.

Recall that a partition of ϕ is a decomposition of ϕ as a sum of positive integers. For example, $\phi = 5$ has 7 partitions since $1+1+1+1+1 = 1+1+1+2 = 1+1+3 = 1+2+2 = 1+4 = 2+3 = 5$. For a partition γ, we let γ_k be the number of times that k appears in the sum, so that $\phi = \sum_{k=1}^{\phi} k\gamma_k$. If γ is the partition corresponding to $1+1+3$, then $\gamma_1 = 2$, $\gamma_3 = 1$, and $\gamma_k = 0$ for k other than 1 or 3. We adopt the convention that Γ_0 consists of the single element γ with $\gamma_k = 0$ for all k.

Equation (1.107) allows us to compute the counts $\tilde{c}_{\tilde{\theta}}(\phi \mid e\,; \mathbf{f}, \mathbf{e})$ in $O(lm + \phi g)$ operations, where g is the number of partitions of ϕ. Although g grows with ϕ like $(4\sqrt{3}\phi)^{-1} \exp \pi\sqrt{2\phi/3}$ [60], it is manageably small for small ϕ. For example, $\phi = 10$ has 42 partitions.

Proof of Formula (1.107)

Introduce the generating functional

$$G(x \mid e, \mathbf{f}, \mathbf{e}) = \sum_{\phi=0}^{\infty} \tilde{c}_{\tilde{\theta}}(\phi \mid e\,; \mathbf{f}, \mathbf{e}) x^\phi$$

where x is an indeterminant. Then

$$\begin{aligned}
G(x \mid e, \mathbf{f}, \mathbf{e}) &= \sum_{\phi=0}^{\infty} \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} \tilde{p}_{\tilde{\theta}}(a_j \mid j, \mathbf{f}, \mathbf{e}) \sum_{i=1}^{l} \delta(e, e_i) \delta(\phi, \phi_i) x^{\phi} && (1.111)\\
&= \sum_{i=1}^{l} \delta(e, e_i) \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} \tilde{p}_{\tilde{\theta}}(a_j \mid j, \mathbf{f}, \mathbf{e}) x^{\phi_i} && (1.112)\\
&= \sum_{i=1}^{l} \delta(e, e_i) \sum_{a_1=0}^{l} \cdots \sum_{a_m=0}^{l} \prod_{j=1}^{m} \tilde{p}_{\tilde{\theta}}(a_j \mid j, \mathbf{f}, \mathbf{e}) x^{\delta(i, a_j)} && (1.113)\\
&= \sum_{i=1}^{l} \delta(e, e_i) \prod_{j=1}^{m} \sum_{a=0}^{l} \tilde{p}_{\tilde{\theta}}(a \mid j, \mathbf{f}, \mathbf{e}) x^{\delta(i, a)} && (1.114)\\
&= \sum_{i=1}^{l} \delta(e, e_i) \prod_{j=1}^{m} (1 - \tilde{p}_{\tilde{\theta}}(i \mid j, \mathbf{f}, \mathbf{e}) + x \tilde{p}_{\tilde{\theta}}(i \mid j, \mathbf{f}, \mathbf{e})) && (1.115)\\
&= \sum_{i=1}^{l} \delta(e, e_i) \prod_{j=1}^{m} (1 - \tilde{p}_{\tilde{\theta}}(i \mid j, \mathbf{f}, \mathbf{e})) \prod_{j=1}^{m} (1 + \beta_{ij}(\mathbf{f}, \mathbf{e}) x). && (1.116)
\end{aligned}$$

To obtain Equation (1.112), rearrange the order of summation and sum over ϕ to eliminate the δ-function of ϕ. To obtain Equation (1.113), note that $\phi_i = \sum_{j=1}^{m} \delta(i, a_j)$ and so $x^{\phi_i} = \prod_{j=1}^{m} x^{\delta(i, a_j)}$. To obtain Equation (1.114), interchange the order of the sums on a_j and the product on j. To obtain Equation (1.115), note that in the sum on a, the only term for which the power of x is nonzero is the one for which $a = i$.

Now note that for any indeterminants $x, y_1, y_2, \cdots, y_m$,

$$\prod_{j=1}^{m} (1 + y_j x) = \sum_{\phi=0}^{m} x^{\phi} \sum_{\gamma \in \Gamma_{\phi}} \prod_{k=1}^{\phi} \frac{{z_k}^{\gamma_k}}{\gamma_k !}, \tag{1.117}$$

$$\text{where } z_k = \frac{(-1)^{k+1}}{k} \sum_{j=1}^{m} (y_j)^k. \tag{1.118}$$

This follows from the calculation

$$\begin{aligned}
\prod_{j=1}^{m} (1 + y_j x) &= \exp \sum_{j=1}^{m} \log(1 + y_j x) = \exp \sum_{j=1}^{m} \sum_{k=1}^{\infty} \frac{(-1)^{k+1} (y_j x)^k}{k} && (1.119)\\
&= \exp \sum_{k=1}^{\infty} z_k x^k = \sum_{n=0}^{\infty} \frac{1}{n!} (\sum_{k=1}^{\infty} z_k x^k)^n && (1.120)\\
&= \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{\gamma_1} \sum_{\gamma_2} \cdots \begin{pmatrix} n \\ \gamma_1 \gamma_2 \cdots \end{pmatrix} \prod_{k=1}^{\infty} (z_k x^k)^{\gamma_k} = \sum_{\phi=0}^{\infty} x^{\phi} \sum_{\gamma \in \Gamma_{\phi}} \prod_{k=1}^{\infty} \frac{{z_k}^{\gamma_k}}{\gamma_k !}. && (1.121)
\end{aligned}$$

The reader will notice that the left-hand side of Equation (1.119) involves only powers of x up to m, while the subsequent Equations (1.120) - (1.121) involve all powers of x. This is because the z_k are not algebraically independent. In fact, for $\phi > m$, the coefficient of x^{ϕ} on the right-hand side of Equation (1.121) must be zero. It follows that we can express z_{ϕ} as a polynomial in z_k, $k = 1, 2, \cdots, m$.

Using Equation (1.117) we can identify the coefficient of x^{ϕ} in Equation (1.116). We obtain Equation (1.107) by combining Equations (1.116), (1.117), and the Definitions (1.108) - (1.110)

1.2.6 Model 4

Parameters

$t(f \mid e)$	translation probabilities
$n(\phi \mid e)$	fertility probabilities
p_0, p_1	fertility probabilities for e_0
$d_1(\Delta j \mid \mathcal{A}, \mathcal{B})$	distortion probabilities for the first word of a tablet
$d_{>1}(\Delta j \mid \mathcal{B})$	distortion probabilities for the other words of a tablet

Here Δj is an integer; $\mathcal{A}$ is an English class; and $\mathcal{B}$ is a French class.

General Formulae

$$\mathrm{P}_\theta(\tau, \pi \mid \mathbf{e}) = \mathrm{P}_\theta(\phi \mid \mathbf{e})\mathrm{P}_\theta(\tau \mid \phi, \mathbf{e})\mathrm{P}_\theta(\pi \mid \tau, \phi, \mathbf{e}) \tag{1.122}$$

$$\mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = \sum_{(\tau,\pi)\in\langle \mathbf{f},\mathbf{a}\rangle} \mathrm{P}_\theta(\tau, \pi \mid \mathbf{e}) \tag{1.123}$$

Assumptions

$$\mathrm{P}_\theta(\phi \mid \mathbf{e}) = n_0(\phi_0 \mid \sum_{i=1}^{l} \phi_i) \prod_{i=1}^{l} n(\phi_i \mid e_i) \tag{1.124}$$

$$\mathrm{P}_\theta(\tau \mid \phi, \mathbf{e}) = \prod_{i=0}^{l} \prod_{k=1}^{\phi_i} t(\tau_{ik} \mid e_i) \tag{1.125}$$

$$\mathrm{P}_\theta(\pi \mid \tau, \phi, \mathbf{e}) = \frac{1}{\phi_0!} \prod_{i=1}^{l} \prod_{k=1}^{\phi_i} p_{ik}(\pi_{ik}) \tag{1.126}$$

where

$$n_0(\phi_0 \mid m') = \begin{pmatrix} m' \\ \phi_0 \end{pmatrix} {p_0}^{m'-\phi_0} {p_1}^{\phi_0}, \tag{1.127}$$

$$p_{ik}(j) = \begin{cases} d_1(j - c_{\rho_i} \mid \mathcal{A}(e_{\rho_i}), \mathcal{B}(\tau_{i1})) & \text{if } k = 1 \\ d_{>1}(j - \pi_{ik-1} \mid \mathcal{B}(\tau_{ik})) & \text{if } k > 1 \end{cases}. \tag{1.128}$$

In Equation (1.128), ρ_i is the first position to the left of i for which $\phi_i > 0$, and c_ρ is the ceiling of the average position of the words of τ_ρ:

$$\rho_i = \max_{i'<i}\{i' : \phi_{i'} > 0\}, \qquad c_\rho = \lceil {\phi_\rho}^{-1} \sum_{k=1}^{\phi_\rho} \pi_{\rho k} \rceil. \tag{1.129}$$

This model is deficient, since

$$\mathrm{P}_\theta(\mathit{failure} \mid \tau, \phi, \mathbf{e}) \equiv 1 - \sum_{\pi} \mathrm{P}_\theta(\pi \mid \tau, \phi, \mathbf{e}) > 0. \tag{1.130}$$

Note that Equations (1.124), (1.125), and (1.127) are identical to the corresponding formulae (1.92), (1.93), and (1.95) for Model 3.

Generation

Equations (1.122) - (1.126) describe the following process for producing $\mathbf{f}$ or *failure* from $\mathbf{e}$:

1.-4. Choose a tableau τ by following Steps $1 - 4$ for Model 3.

5. For each $i = 1, 2, \cdots, l$ and each $k = 1, 2, \cdots, \phi_i$ choose a position π_{ik} as follows.

If $k = 1$ then choose π_{i1} according to the distribution $d_1(\pi_{i1} - c_{\rho_i} \mid \mathcal{A}(e_{\rho_i}), \mathcal{B}(\tau_{i1}))$.

If $k > 1$ then choose π_{ik} greater than π_{ik-1} according to the distribution $d_{>1}(\pi_{ik} - \pi_{ik-1} \mid \mathcal{B}(\tau_{ik}))$.

6.-8. Finish generating $\mathbf{f}$ by following Steps 6 – 8 for Model 3.

1.2.7 Model 5

Parameters

$t(f \mid e)$	translation probabilities
$n(\phi \mid e)$	fertility probabilities
p_0, p_1	fertility probabilities for e_0
$d_1(\Delta j \mid \mathcal{B}, v)$	distortion probabilities for the first word of a tablet
$d_{>1}(\Delta j \mid \mathcal{B}, v)$	distortion probabilities for the other words of a tablet

Here $v = 1, 2, \cdots, m$.

General Formulae

$$\mathrm{P}_\theta(\tau, \pi \mid \mathbf{e}) = \mathrm{P}_\theta(\phi \mid \mathbf{e})\mathrm{P}_\theta(\tau \mid \phi, \mathbf{e})\mathrm{P}_\theta(\pi \mid \tau, \phi, \mathbf{e}) \tag{1.131}$$

$$\mathrm{P}_\theta(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = \sum_{(\tau,\pi)\in\langle\mathbf{f},\mathbf{a}\rangle} \mathrm{P}_\theta(\tau, \pi \mid \mathbf{e}) \tag{1.132}$$

Assumptions

$$\mathrm{P}_\theta(\phi \mid \mathbf{e}) = n_0(\phi_0 \mid \sum_{i=1}^{l} \phi_i) \prod_{i=1}^{l} n(\phi_i \mid e_i) \tag{1.133}$$

$$\mathrm{P}_\theta(\tau \mid \phi, \mathbf{e}) = \prod_{i=0}^{l} \prod_{k=1}^{\phi_i} t(\tau_{ik} \mid e_i) \tag{1.134}$$

$$\mathrm{P}_\theta(\pi \mid \tau, \phi, \mathbf{e}) = \frac{1}{\phi_0!} \prod_{i=1}^{l} \prod_{k=1}^{\phi_i} p_{ik}(\pi_{ik}) \tag{1.135}$$

where

$$n_0(\phi_0 \mid m') = \binom{m'}{\phi_0} {p_0}^{m'-\phi_0} {p_1}^{\phi_0}, \tag{1.136}$$

$$\tag{1.137}$$

$$p_{ik}(j) = \epsilon_{ik}(j) \begin{cases} d_1(v_{i1}(j) - v_{i1}(c_{\rho_i}) \mid \mathcal{B}(\tau_{i1}), v_{i1}(m) - \phi_i + k) & \text{if } k = 1 \\ d_{>1}(v_{ik}(j) - v_{ik}(\pi_{ik-1}) \mid \mathcal{B}(\tau_{ik}), v_{ik}(m) - v_{ik}(\pi_{ik-1}) - \phi_i + k) & \text{if } k > 1 \end{cases} \tag{1.138}$$

In Equation (1.138), ρ_i is the first position to the left of i which has a non-zero fertility; and c_ρ is the ceiling of the average position of the words of tablet ρ (see Equation (1.129)). Also, $\epsilon_{ik}(j)$ is 1 if position j is vacant after all the words of tablets $i' < i$ and the first $k - 1$ words of tablet i have been placed, and 0 otherwise. $v_{ik}(j)$ is the number of vacancies not to the right of j at this time: $v_{ik}(j) = \sum_{j' \leq j} \epsilon_{ik}(j')$.

This model is not deficient. Note that Equations (1.133), (1.134), and (1.137) are identical to the corresponding formulae for Model 3.

Generation

Equations (1.131) - (1.135) describe the following process for producing $\mathbf{f}$ from $\mathbf{e}$:

1.-4. Choose a tableau τ by following Steps $1-4$ for Model 3.

5. For each $i = 1, 2, \cdots, l$ and each $k = 1, 2, \cdots, \phi_i$ choose a position π_{ik} as follows:

   If $k = 1$ then choose a vacant position π_{i1} according to the distribution
   $d_1(v_{i1}(\pi_{i1}) - v_{i1}(c_{\rho_i}) \mid \mathcal{B}(\tau_{i1}), v_{i1}(m) - \phi_i + k)$.

   If $k > 1$ then choose a vacant position π_{ik} greater than π_{ik-1} according to the distribution
   $d_{>1}(v_{ik}(\pi_{ik}) - v_{ik}(\pi_{ik-1}) \mid \mathcal{B}(\tau_{ik}), v_{ik}(m) - v_{ik}(\pi_{ik-1}) - \phi_i + k)$.

6.-8. Finish generating $\mathbf{f}$ by following Steps $6-8$ for Model 3.