[Human Language Technology, Plainsboro, 1994]

The Candide System for Machine Translation

Adam L. Berger, Peter F. Brown[*] Stephen A. Della Pietra, Vincent J. Della Pietra, John R. Gillett, John D. Lafferty, Robert L. Mercer[*] Harry Printz, Luboš Ureš

IBM Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

ABSTRACT

We present an overview of Candide, a system for automatic translation of French text to English text. Candide uses methods of information theory and statistics to develop a probability model of the translation process. This model, which is made to accord as closely as possible with a large body of French and English sentence pairs, is then used to generate English translations of previously unseen French sentences. This paper provides a tutorial in these methods, discussions of the training and operation of the system, and a summary of test results.

1. Introduction

Candide is an experimental computer program, now in its fifth year of development at IBM, for translation of French text to English text. Our goal is to perform fully-automatic, high-quality text-to-text translation. However, because we are still far from achieving this goal, the program can be used in both fully-automatic and translator's-assistant modes.

Our approach is founded upon the statistical analysis of language. Our chief tools are the source-channel model of communication, parametric probability models of language and translation, and an assortment of numerical algorithms for training such models from examples. This paper presents elementary expositions of each of these ideas, and explains how they have been assembled to produce Candide.

In Section 2 we introduce the necessary ideas from information theory and statistics. The reader is assumed to know elementary probability theory at the level of [1]. In Sections 3 and 4 we discuss our language and translation models. In Section 5 we describe the operation of Candide as it translates a French document. In Section 6 we present results of our internal evaluations and the ARPA Machine Translation Project evaluations. Section 7 is a summary and conclusion.

2. Statistical Translation

Consider the problem of translating French text to English text. Given a French sentence **f**, we imagine that it was originally rendered as an equivalent English sentence **e**. To obtain the French, the English was transmitted over a noisy communication channel, which has the curious property that English sentences sent into it emerge as their French translations. The central assumption of Candide's design is that the characteristics of this channel can be determined experimentally, and expressed mathematically.

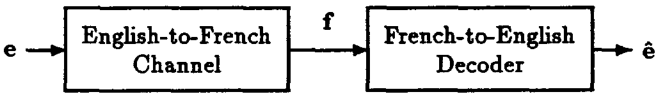

Figure 1: The Source–Channel Formalism of Translation. Here **f** is the French text to be translated, **e** is the putative original English rendering, and **ê** is the English translation.

This formalism can be exploited to yield French-to-English translations as follows. Let us write $\Pr(\mathbf{e} \mid \mathbf{f})$ for the probability that **e** was the original English rendering of the French **f**. Given a French sentence **f**, the problem of automatic translation reduces to finding the English sentence that maximizes $\Pr(\mathbf{e} \mid \mathbf{f})$. That is, we seek $\hat{\mathbf{e}} = \operatorname{argmax}_{\mathbf{e}} \Pr(\mathbf{e} \mid \mathbf{f})$.

By virtue of Bayes' Theorem, we have

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{e} \mid \mathbf{f}) = \operatorname*{argmax}_{\mathbf{e}} \Pr(\mathbf{f} \mid \mathbf{e}) \Pr(\mathbf{e}) \qquad (1)$$

The term $\Pr(\mathbf{f} \mid \mathbf{e})$ models the probability that **f** emerges from the channel when **e** is its input. We call this function the *translation model*; its domain is all pairs $\langle \mathbf{f}, \mathbf{e} \rangle$ of French and English word-strings. The term $\Pr(\mathbf{e})$ models the *a priori* probability that **e** was supplied as the channel input. We call this function the *language model*. Each of these factors—the translation model and the language model—independently produces a score for a candidate English translation **e**. The translation model ensures that the words of **e** express the ideas of **f**, and the language model ensures that **e** is a grammatical sentence. Candide selects as its translation the **e** that maximizes their product.

This discussion begs two important questions. First, where do the models $\Pr(\mathbf{f} \mid \mathbf{e})$ and $\Pr(\mathbf{e})$ come from? Second, even if we can get our hands on them, how can we search the set of all English strings to find $\hat{\mathbf{e}}$? These questions are addressed in the next two sections.

2.1. Probability Models

We begin with a brief detour into probability theory. A *probability model* is a mathematical formula that purports to express the chance of some observation. A *parametric model* is a probability model with adjustable parameters, which can be changed to make the model better match some body of data.

Let us write **c** for a body of data to be modeled, and θ for a vector of parameters. The quantity $\Pr_\theta(\mathbf{c})$, computed according to some formula involving **c** and θ, is called the likelihood

[*] Current address: Renaissance Technologies, Stony Brook, NY

of c. It is the model's assignment of probability to the observation sequence c, according to the current parameter values θ. Typically the formula for the likelihood includes some constraints on the elements of θ to ensure that $\Pr_\theta(\mathbf{c})$ really is a probability distribution—that is, it is always a real value in $[0, 1]$, and for fixed θ the sum $\sum_{\mathbf{c}} \Pr_\theta(\mathbf{c})$ over all possible c vectors is 1.

Consider the problem of *training* this parametric model to the data c; that is, adjusting the θ to maximize $\Pr_\theta(\mathbf{c})$. Finding the maximizing θ is an exercise in constrained optimization. If the expression for $\Pr_\theta(\mathbf{c})$ is of a suitable (simple) form, the maximizing parameter vector $\hat{\theta}$ can be solved for directly.

The key elements of this problem are

- a vector θ of adjustable parameters,
- constraints on these parameters to ensure that we have a model,
- a vector c of observations, and
- the adjustment of θ, subject to constraints, to maximize the likelihood $\Pr_\theta(\mathbf{c})$.

We often seek more than a probability model of some observed data c. There may be some hidden statistics **h**, which are related to c, but which are never directly revealed; in general **h** itself is restricted to some set $\mathcal{H}$ of admissible values. For instance, c may be a large corpus of grammatical text, and **h** an assignment of parts-of-speech to each of its words.

In such cases, we proceed as follows. First we write down a parametric model $\Pr_\theta(\mathbf{c}, \mathbf{h})$. Then we attempt to adjust the parameter vector θ to maximize the likelihood $\Pr_\theta(\mathbf{c})$, where this latter is obtained as the sum $\sum_{\mathbf{h}\in\mathcal{H}} \Pr_\theta(\mathbf{c}, \mathbf{h})$.

Unfortunately, when we attempt to solve this more complicated problem, we often discover that we cannot find a closed-form solution for $\hat{\theta}$. Instead we obtain formulae that express each of the desired parameters in terms of all the others, and also in terms of the observation vector c.

Nevertheless, we can frequently apply an iterative technique called the *Expectation-Maximization* or *EM Algorithm*; this is a recipe for computing a sequence θ_1, θ_2, ... of parameter vectors. It can be shown [2] that under suitable conditions, each iteration of the algorithm is guaranteed to produce a better model of the training vector c; that is,

$$\Pr_{\theta_{i+1}}(\mathbf{c}) \geq \Pr_{\theta_i}(\mathbf{c}), \qquad (2)$$

with strict inequality everywhere except at stationary points of $\Pr_\theta(\mathbf{c})$. When we adjust the model's parameters this way, we say it has been *EM-trained*.

Training a model with hidden statistics is just like training one that lacks them, except that it is not possible to find a maximizing $\hat{\theta}$ in just one go. Training is now an iterative process, involving repeated passes over the observation vector. Each pass yields an improved model of that data.

Now we relate these methods to the problem at hand, which is to develop a translation model $\Pr(\mathbf{f} \mid \mathbf{e})$, and a language model $\Pr(\mathbf{e})$. Consider the translation model. As any first-year language student knows, word-for-word translation of English to French does not work. The dictionary equivalents of the English words can move forward or backward in the sentence, they may disappear completely, and new French words may appear to arise spontaneously.

Guided by this observation, our approach has been to write down an enormous parametric expression, $\Pr_\theta(\mathbf{f} \mid \mathbf{e})$, for the translation model. To give the reader some idea of the scale of the computation, there is a parameter, $t(f|e)$, for the probability that any given English word e will translate as any given French word f. There are parameters for the probability that any f may arise spontaneously, and that any e may simply disappear. There are parameters that words may move forward or backward 1, 2, 3, ... positions. And so on.

We use a similar approach to write an expression for $\Pr_\theta(\mathbf{e})$. In this case the parameters express things like the probability that a word e_i may appear in a sentence after some word sequence $e_1 e_2 \ldots e_{i-1}$. In general, the parameters are of the form $\Pr(e_i|\mathbf{v})$, where the vector v is a combination of observable statistics like the identities of nearby words, and hidden statistics like the grammatical structure of the sentence. We refer to v as a *history*, from which we predict e_i.

The parameter values of both models are determined by EM training. For the translation model, the training data consists of English–French sentence pairs $\langle \mathbf{e}, \mathbf{f} \rangle$, where e and f are translations of one another. For the language model, it consists exclusively of English text.

2.2. Decoding

We do not actually search the infinite set of all English word strings to find the ê that maximizes equation (1). Even if we restricted ourselves to word strings of length k or less, for any realistic length and English vocabulary $\mathcal{E}$, this is far too large a set to search exhaustively. Instead we adapt the well-known stack decoding algorithm [5] of speech recognition. Though we will say more about decoding in Section 6 below, most of our research effort has been devoted to the two modeling problems.

This is not without reason. The translation scheme we have just described can fail in only two ways. The first way is a search error, which means that our decoding procedure did not yield the ê that maximizes $\Pr(\mathbf{f} \mid \mathbf{e})\Pr(\mathbf{e})$. The second way is a modeling error, which means that the best English translation, as supplied by a competent human, did not maximize this same product. Our tests show that only 5% of our system's errors are search errors.

3. Language Modeling

Let e be a string of English words $e_1 \ldots e_\ell$. A language model $\Pr(\mathbf{e})$ gives the probability that e would appear in grammatical English text.

By the laws of conditional probability we may write

$$\begin{aligned} \Pr(\mathbf{e}) &= \Pr(e_1 \ldots e_\ell) \\ &= \Pr(e_1)\Pr(e_2|e_1)\Pr(e_3|e_1e_2)\cdots\Pr(e_\ell|e_1\ldots e_{\ell-1}). \end{aligned}$$

Given this decomposition the language modeler's job is to estimate each of the ℓ distributions on the right hand side.

If $|\mathcal{E}|$ is the size of the English vocabulary, then the number of different histories $e_1 \ldots e_{k-1}$ in the kth conditional grows as $|\mathcal{E}|^{k-1}$. This presents problems both in practice and in principle—the former because we don't have enough storage to write down all the different histories, the latter because even if we could, any one history would be exceedingly rare, making it impossible to estimate probabilities accurately.

For these reasons, Candide has used the so-called *trigram model* as its workhorse. In this model, we use the approximation

$$\Pr(e_k \mid e_1 \ldots e_{k-1}) \approx \Pr(e_k \mid e_{k-2} e_{k-1})$$

for each term on the right hand side above. That is, we limit the history to two words. Each triple $\langle e_{k-2} e_{k-1} e_k \rangle$ is called a *trigram*.

It remains to estimate the $\Pr(e_k | e_{k-2} e_{k-1})$. One solution is to use maximum-likelihood trigram probabilities, $T(e_k | e_{k-2} e_{k-1})$. These are obtained by scanning the training corpus $\mathbf{c}$, counting the incidence of each trigram, and using these counts to form the appropriate conditional estimates.

But even for this modest history size, we frequently encounter trigrams during translation that do not appear during training. This is not surprising, since there are $|\mathcal{E}|^3 = 1.773 \times 10^{16}$ possible different trigrams, yet we can encounter no more than $|\mathbf{c}|$ of them during training. There are 75,349,888 distinct trigrams in our training corpus, of which 53,737,350 occur exactly once.

For this reason, we employ the technique of *deleted interpolation* [6]: we express $\Pr(e_k | e_{k-2} e_{k-1})$ as a linear combination of the trigram probability $T(e_k | e_{k-2} e_{k-1})$, the bigram probability $B(e_k | e_{k-1})$, the unigram probability $U(e_k)$, and the uniform probability $1/|\mathcal{E}|$. The distributions B and U are obtained by counting the incidence of bigrams and unigrams in the same training corpus $\mathbf{c}$. But there are fewer distinct bigrams, so we have a higher chance of seeing any given one in our training data, and a still higher chance of seeing any given unigram. The resulting formula for $\Pr(e_k | e_{k-2} e_{k-1})$ is called the *smoothed trigram model.*

Even the smoothed trigram model leaves much to be desired, since it does not account for semantic and syntactic dependencies among words that do not lie within the same trigram. This has led us to use a *link grammar* model. This is a trainable, probabilistic grammar that attempts to capture all the information present in the trigram model, and also to make the long-range connections among words needed to advance beyond it. Link grammars are discussed in detail in [7].

4. Translation Modeling

This section describes the elements of our translation model, $\Pr(\mathbf{f} \mid \mathbf{e})$. We have two distinct translation models, both described here: an EM-trained model, and a maximum-entropy model.

As we explain in Section 4.2 below, the EM-trained model is developed through a succession of five provisional models. Before we describe them, we introduce the notion of *alignment.*

4.1. Alignment

Consider a pair of French and English sentences $\langle \mathbf{e}, \mathbf{f} \rangle$ that are translations of one another. Although we argued above that word-for-word translation will not work to develop $\mathbf{f}$ from $\mathbf{e}$, it is clear that there is some relation between the individual words of the two sentences. A typical assignment of relations is depicted in Figure 2.

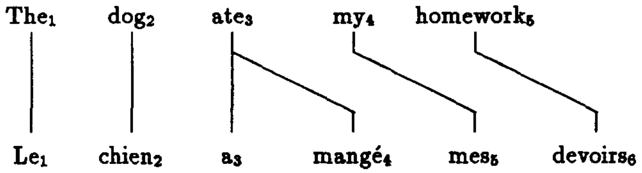

Figure 2: Alignment of a French–English Sentence Pair. The subscripts give the position of each word in the sentence.

We call such a set of connections between sentences an *alignment.* Formally we express it as a set $\mathbf{a}$ of pairs $\langle j, i \rangle$, where each pair stands for a connection between the jth word of $\mathbf{f}$ and the ith word of $\mathbf{e}$. Our intention is to connect f_j and e_i when e_i was one of the words expressing in English the concept that f_j (possibly along with other words of $\mathbf{f}$) expresses in French. In its most general form, an alignment may consist of any set $\mathbf{a}$ of $\langle j, i \rangle$ pairs. But for simplicity, we restrict ourselves to alignments in which each French word is connected to a unique English word.

We cannot hope to discover alignments with certainty. Our strategy is to train a parametric model for the joint distribution $\Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{e})$, where the alignment $\mathbf{a}$ is hidden. In principle, the desired conditional $\Pr(\mathbf{f} \mid \mathbf{e})$ may then be obtained as $\sum_{\mathbf{a}} \Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{e})$, where the sum is taken over all possible alignments of $\mathbf{e}$ and $\mathbf{f}$. In practice this is possible only for our first two models. For the remaining models, we approximate $\Pr(\mathbf{f} \mid \mathbf{e})$ as follows. During training, we find the single most probable alignment $\hat{\mathbf{a}}$, and sum $\Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{e})$ over a small neighborhood of $\hat{\mathbf{a}}$. During decoding, we simply use $\Pr(\mathbf{f}, \hat{\mathbf{a}} \mid \mathbf{e})$.

4.2. EM-Trained Models

We now sketch the structure of five models of increasing complexity, the last of which is our EM-trained translation model. For an in-depth treatment, the reader is referred to [3].

1. Word Translation This is our simplest model, intended to discover probable individual-word translations. The free parameters of this model are word translation probabilities $t(f_j \mid e_i)$. Each of these parameters is initialized to $1/|\mathcal{F}|$, where $\mathcal{F}$ is our French vocabulary. Thus we make no initial assumptions about appropriate French–English word pairings. The iterative training procedure automatically finds appropriate translations, and assigns them high probability.

2. Local Alignment To make our model more realistic, we introduce an alignment variable a_j for each position j of $\mathbf{f}$; a_j is the position in $\mathbf{e}$ to which the jth word of $\mathbf{f}$ is aligned. (French words that appear to arise spontaneously are said to align to the *null word*, in position 0 of $\mathbf{e}$.) Formally, we insert a parameter $\Pr(a_j \mid j, m, l)$ into our expression for $\Pr(\mathbf{f}, \mathbf{a} \mid \mathbf{e})$. This expresses the probability that *position* j in an arbitrary French sentence of length m is aligned with *position* a_j in any English sentence of length l that is its translation. The identities of the words in these positions do not influence the alignment probabilities.

3. Fertilities As we observed earlier, a single English word may yield 0, 1 or more French words, for instance as when *not* translates to *ne...pas*. This idea is implicit in our notion of alignment, but not explicitly related to word identities. To capture this phenomenon explicitly, this model introduces the notion of *fertility*. The fertility $\phi(e_i)$ is the number of French words in $\mathbf{f}$ that e_i generates in translation. Fertility is incorporated into this model through the parameters $\phi(n|e_i)$, the probability that $\phi(e_i)$ equals n.

4. Class-Based Alignment In the preceding model, though the fertilities are conditioned upon word identities, the alignment parameters are not. We have already pointed out how unrealistic this is, since it aligns *positions*in the $\langle \mathbf{e}, \mathbf{f} \rangle$ pair with no regard for the words found there. This model remedies the problem by expressing alignments in terms of parameters that depend upon the *classes* of words that lie at the aligned positions. Each word f in our French vocabulary $\mathcal{F}$ is placed in one of approximately 50 classes; likewise for each e in the English vocabulary $\mathcal{E}$. The assignment of words to classes is made automatically through another statistical training procedure [3].

5. Non-Deficient Alignment The preceding two models suffer from a problem we call *deficiency*: they assign non-zero probability to "alignments" that do not correspond to strings of French words at all. For instance, two French words may be assigned to lie at the same position in the sentence. Words may be placed before the start of the sentence, or after its end. This model eliminates such spurious alignments.

These five models are trained in succession on the same data, with the final parameter values of one model serving as the starting point for the next. For the current version of Candide, we used a corpus of 2,205,733 English–French sentence pairs, drawn mostly from the Hansards, which are the proceedings of the Canadian Parliament. The entire computation took a total of approximately 3600 processor-hours distributed over fifteen IBM Model 530H POWERstations.

The reader may be wondering why we have five translation models instead of one. This is because the EM algorithm, though guaranteed to converge to a local maximum, need not converge to a global one. A weakness of the algorithm is that it may yield a parameter vector $\hat{\theta}$ that is indeed a local maximum, but which does not model the data well.

It so happens though that model 1 has a special form that ensures that EM training *is* guaranteed to converge to a global maximum. By using model 1's final parameter vector as the initial vector for model 2, we are assured that we are at a reasonably good starting point for training the latter. By extension of this argument, we proceed through the training of each model in succession, with some confidence that each model's starting point is a good one.

4.3. Context Sensitive Models

All of the preceding translation models make one important simplification: each English word acts independently of all the others in the sentence to generate the French words to which it is aligned. But it is easy to convince oneself that this approach is inadequate; clearly *run* will translate differently in *Let's run the program!* and *Let's run the race!*. Intuitively, we would like to make the translation of a word depend upon context in which it appears.

For this reason, we have constructed translation models that take context into account. Our instinct is to make the translation of a word depend upon its neighbors, say writing $t(f_j \mid e_i \; e_{i\pm1} \ldots)$ for the word-translation probabilities. But this is impractical, because of the same difficulties that confront language models with long histories.

To overcome this, we employ a technique—maximum-entropy modeling—that deals with small chosen subsets of a potentially large number of conditioning variables. We begin with a large set $Q = \{b_1(f, e, \mathbf{e}) \; b_2(f, e, \mathbf{e}) \; b_3(f, e, \mathbf{e}) \ldots\}$ of binary-valued functions. Each such function asks some yes/no question about the French word f, the English word e, and the context $\mathbf{e}$ in which e appears.

The training procedure works iteratively to find a small subset $Q' = \{b_{k_1}(f_j, e_i, \mathbf{e}) \;\; b_{k_2}(f_j, e_i, \mathbf{e}) \ldots b_{k_N}(f_j, e_i, \mathbf{e})\}$ that disambiguates the senses of the English word in context. Formally, it develops a distribution $t(f_j \mid e_i \; Q')$ that tells us if f_j is a good translation of e_i in the context $\mathbf{e}$. Since this procedure is costly in computer time, we develop such models only for the 2,000 most common English words. For more information about maximum-entropy modeling, the reader is referred to [4].

5. Analysis–Transfer–Synthesis

Although we try to obtain accurate estimates of the parameters of our translation models by training on a large amount of text, this data is not used as effectively as it might be. For instance, the word-translation probabilities $t(\textit{parle} \mid \textit{speaks})$ and $t(\textit{parlent} \mid \textit{speak})$ must be learned separately, though they express the underlying equivalence of the infinitives *parler* and *to speak*.

For this reason, we have adopted for Candide a variation of the analysis-transfer-synthesis paradigm. In this paradigm, translation takes place not between raw French and English texts, but between intermediate forms of the two languages. Note that because translation is effected between intermediate French and intermediate English, all our models are trained upon intermediate text as well. For training, each $\langle \mathbf{e}, \mathbf{f} \rangle$ pair of our data is subjected to an *analysis* step: the French is rendered into an intermediate French $\mathbf{f}'$, the English into intermediate English $\mathbf{e}'$. The English transformation is constructed to ensure that it is invertible; its inverse, from intermediate English to standard English, is usually called *synthesis*.

The aim of these transformations is three-fold: to suppress lexical variations that conceal regularities between the two languages, to reduce the size of both vocabularies, and to reduce the burden on the alignment model by making coordinating phrases resemble each other as closely as possible with respect to length and word order.

Both the English and the French analysis steps consist of five classes of operations: segmentation, name and number detection, case and spelling correction, morphological analysis, and linguistic normalization. During segmentation, the French is divided (if possible) into shorter phrases that represent distinct concepts. This does not modify the text, but the translation model, used later, respects this division by ignoring alignments that cross segment boundaries.

During name and number detection, numbers and proper names—word strings such as *Ethiopie*, *Grande Bretagne* and *2.84 cm*—are removed from the French text and replaced by generic name and number markers. Removing names and numbers greatly reduces the size of $\mathcal{E}$ and $\mathcal{F}$. The excised texts are translated by rule and kept in a table, to be substituted back into the English sentence during synthesis.

During case and spelling correction, we correct any obvious spelling errors, and suppress the case variations in word spellings that arise from the conventions of English and French typography.

During morphological analysis, we first use a hidden Markov model [8] to assign part-of-speech labels to the French, then use these labels to replace inflected verb forms with their infinitives, preceded by an appropriate tense marker. We also put nouns into singular form and precede them by number markers, and perform a variety of other morphological transformations.

Finally, during linguistic normalization we perform a series of word reorderings, insertions and rewritings intended to regularize each language, and to make the two languages more closely resemble each other. For example, the contractions *au* and *du* are rewritten as *à le* and *de le*. Constructions such as *il y a* and *ne...pas* are replaced with one-word tokens. The English possessive construction is made to resemble French by removing the *'s* or *'* suffix, reordering noun phrases, and inserting an additional token. Thus *my aunt's pen* becomes intermediate English *dummy-article pen 's my aunt*; note the similarity to the French *le stylo de ma tante*.

6. Operation of Candide

In previous sections we have indicated how the parameters of Candide's various models are determined via the EM algorithm and maximum-entropy methods. We now outline the steps involved in the execution of Candide as it translates a French passage into English. The process of translation, divided into analysis, transfer, and synthesis stages, is depicted in Figure 3.

In the analysis stage, the French input string $\mathbf{f}$ is converted into $\mathbf{f}'$, as discussed above. The output of this stage is denoted in Figure 3 as *Intermediate French*.

The transfer stage constitutes the decoding process sketched in Section 2.2 above. Decoding consists of two steps. In the first step, Candide develops a set H^* of candidate decodings, using coarse versions of our translation and language models to select its elements. In the second step, the system expands H^* and rescores the enlarged set using more sophisticated models. We now describe both steps in greater detail.

In the first step, Candide applies a variation of the stack decoding algorithm to generate candidate decodings. Decoding proceeds left-to-right, one intermediate English word at a time. At each stage we maintain a ranked set $H^{(i)}$ of partial hypotheses for the intermediate English $\hat{\mathbf{e}}'$.

In general, the elements of $H^{(i)}$ are *partial* decodings of $\mathbf{f}'$; that is, only the leading i words of $\hat{\mathbf{e}}'$ have been filled in, and these account for only some of the words of $\mathbf{f}'$. To advance the decoding, some elements of $H^{(i)}$ are selected to be extended by one word. The translation and language models work together to generate the $i+1$st word; the resulting partial decodings are ranked; this ranked set is $H^{(i+1)}$. An hypothesis is complete when all words of $\mathbf{f}'$ have been accounted for. Note that while the intermediate English is generated left-to-right, the treatment of intermediate French words does not necessarily proceed left-to-right, due to the word-reordering property of the channel. This is one of the key ways that translation differs from speech—a difference that greatly complicates the decoding process.

The ranking of hypotheses is according to the product $\Pr(\mathbf{f}' \mid \mathbf{e}')\Pr(\mathbf{e}')$. In the interest of speed, and because we must deal with partial rather than complete sentences, we employ the EM-trained translation model and the smoothed trigram language model. The output of this step is a ranked set H^* of the 140 best intermediate English sentences.

During the second step, called *perturbation search*, we enlarge H^* by considering sequences of single-word deletions, insertions or replacements to its elements. Then we rerank the enlarged set using the link grammar language model and the maximum-entropy translation model. The highest-scoring intermediate English sentence that we encounter during perturbation search is the output $\hat{\mathbf{e}}'$ of the transfer stage.

The final stage, synthesis, converts the intermediate English $\hat{\mathbf{e}}'$ into a plain English sentence $\hat{\mathbf{e}}$.

7. Performance

We evaluate our system in two ways: through participation in the ARPA evaluations, and through our own internal tests.

The ARPA evaluation methodology, devised and executed by PRC, is detailed in [9]; we recount it here briefly. ARPA provides us with a set of French passages, which we process in two ways. First, the passages are translated without any human intervention. This is the fully-automatic mode. Second, each of the same passages is translated by two different humans, once with and once without the aid of *Transman*, our translation assistance tool. Transman presents the user with an automated dictionary, a text editor, and parallel views of the French source and the English fully-automatic translation. The passages are ordered in such a way as to suppress the influence of differing levels of translation skill, and the times of all the human translations are recorded.

PRC scores all the resulting English texts for *fluency* and *ad-*

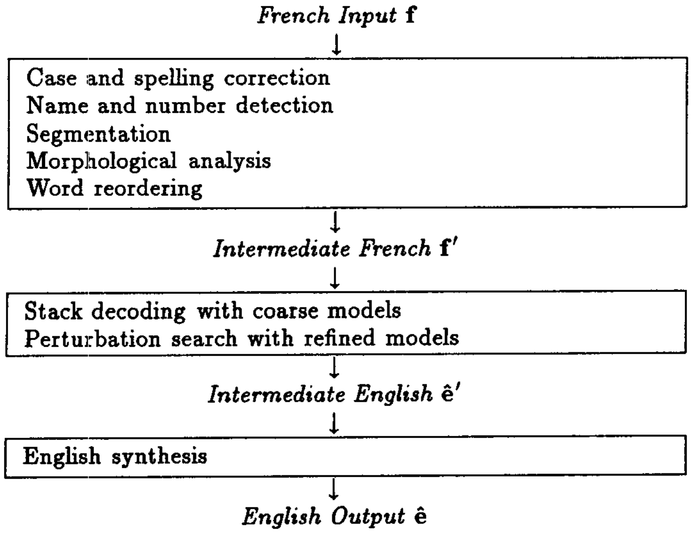

Figure 3: Steps in the Execution of Candide

equacy, reporting these as numbers between 0 and 1. Fluency is intended to measure the well-formedness of translated sentences; adequacy is intended to measure to what extent the meaning of each source text is present in the translations. The advantage afforded by Transman is determined by computing the ratio $t_{Transman}/t_{manual}$ for each passage, where the numerator is the time to translate the passage with Transman's aid, and the denominator is the time for unaided manual translation.

The means of all these statistics are presented in Table 1. As a benchmark, this table includes a line reporting fluency and adequacy results in these tests for Systran, a commercial fully-automatic French–English translation system, considered by some to be the world's best.

Our in-house evaluation methodology consists of fully-automatic translation of 100 sentences of 15 words or less; each translation is judged either *correct* or *incorrect*. These sentences are drawn from the same domain as our training data—the Hansard corpus—but they are of course not sentences that we trained on. Our 1992 system produced 45 correct translations; our 1993 system produced 62 correct translations.

	Fluency		Adequacy		Time Ratio	
	1992	1993	1992	1993	1992	1993
Systran	.466	.540	.686	.743		
Candide	.511	.580	.575	.670		
Transman	.819	.838	.837	.850	.688	.625
Manual		.833		.840		

Table 1: ARPA Evaluation Results. The Systran line reports results for Systran French-to-English fully-automatic translations. The Candide line reports results for our system's fully-automatic translations; the Transman line reports results for our system's machine-assisted translations.

8. Summary

We began with a review of the source–channel formalism of information theory, and how it may be applied to translation. Our approach reduces to formulating and training two parametric probability models: the language model Pr(e), and the translation model Pr(f | e). We described the structure of both models, and how they are trained.

We explained the use of the analysis–transfer–synthesis paradigm, and sketched the system's operation. Finally, we gave performance results for Candide, in both its human-assisted and fully-automatic operating modes.

In our opinion, the most promising avenues for exploration are: the continued elaboration of the link grammar language model, more sophisticated translation models, the maximum-entropy modeling technique, and a more systematic approach to French and English morphological and syntactic analysis.

References

1. Allen, Arnold O. *Probability, Statistics and Queueing Theory*, Academic Press, New York, NY, 1978.

2. Baum, L. E. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, **3**, 1972, pp 1–8.

3. Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, Robert L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics* **19**(2), June 1993, pp 263–311.

4. Jaynes, E. T. Notes on present status and future prospects. *Maximum Entropy and Bayesian Methods*, W. T. Grandy and L. H. Schick, eds. Kluwer Academic Press, 1990, pp 1–13.

5. Jelinek, Frederick. A fast sequential decoding algorithm using a stack. *IBM Journal of Research and Development*, **13**, November 1969, pp 675–685.

6. Jelinek, F., R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings, Workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands, 1980.

7. Lafferty, John, Daniel Sleator, Davy Temperly. Grammatical trigrams: a probabilistic model of link grammar. *Proceedings of the 1992 AAAI Fall Symposium on Probabilistic Approaches to Natural Language.*

8. Merialdo, Bernard. Tagging text with a probabilistic model. *Proceedings of the IBM Natural Language ITL*, Paris, France, 1990, pp 161-172.

9. White, John S., Theresa A. O'Connell, Lynn M. Carlson. Evaluation of machine translation. In *Human Language Technology*, Morgan Kaufman Publishers, 1993, pp 206–210.