$\rho < 1$ we prefer fewer (longer) phrases, and if $\rho > 1$ we prefer more (shorter) phrases.

In practice, if the model has the choice of using a longer phrase translation, it tends to use it. This is preferable because longer phrases include more context. Of course, longer phrases are less frequent and hence less statistically reliable, but we have addressed this problem with lexical weighting, which discounts bad phrase pairs.

Several researchers have explored other avenues for modeling phrase segmentation. One tempting approach is to prefer segmentation along linguistically motivated constituent boundaries. However, this has not been shown to be useful. There are many examples of good phrase translations that cross such boundaries. Recall our initial example, where we produced the English *fun with the*. Such a segmentation within a prepositional phrase is unintuitive from a linguistic point of view. However, it has been shown to be beneficial in phrase-based models. In this case, the translation of the preposition depends more strongly on the preceding noun *fun* than on the following words. Learning such a non-constituent phrase pair captures the dependency.

5.3.6 Phrase Translation as a Classification Problem

One inherent concern with the decomposition of the translation of a sentence into the independent translation of phrases is the loss of context when making translation predictions. The translation of an ambiguous word may depend on other words in the sentence. For instance, the translation of *bat* will be different, if the word *pitcher* or *cave* occurs in the same sentence. Findings in research on word sense disambiguation have shown that a context window of, say, 50–100 words surrounding the ambiguous word may be helpful in determining its meaning.

How can we include such **contextual information** in the phrase translation table? Borrowing ideas from the field of machine learning, we can view phrase translation as a **classification** task. An input phrase translates into one of a fixed set of possible output phrases. So, we have to classify a phrase into its correct translation category.

contextual information

classification

One method for building such phrase translation classifiers is maximum entropy modeling, which we describe in detail in Section 9.2.4 of this book. This method allows the conditioning of phrase translation not only on the input phrase but on any feature that may be found in the training data. Typically, for computational reasons, we restrict such models to properties of the input sentence.

5.4 Extensions to the Reordering Model

So far, we have introduced only a relatively simple reordering model for phrase-based statistical machine translation. Like the reordering model for the IBM Models 3–5, it is based on movement distance. However, it is not conditioned on words or word classes and not even trained on the data.

In this section, we will take a closer look at the issue of reordering, discuss restrictions on reordering, and describe a lexicalized reordering model.

5.4.1 Reordering Limits

Reordering is one of the hardest problems in machine translation. However, it manifests itself differently for different language pairs. Much recent statistical machine translation research has been driven by language pairs such as Chinese–English, Arabic–English and French–English. While these pairs represent a diverse set of input languages, they have one thing in common: restricting reordering to short local movements is sufficient for the translation of most sentences.

Contrast this to the situation when translating from Japanese or German. These languages have a different syntactic structure from English. Most importantly, for the task of reordering, they are (in the case of German mostly) verb-final, meaning the verb occurs at the end of the sentence. The movement of the verb from the end of the sentence to the position just after the subject at the beginning of the sentence is often a move over a large number of words, which would be penalized heavily by the relative distance reordering model.

Our reordering model generally punishes movement. It is up to the language model to justify the placement of words in a different order in the output. For the local changes required when translating from, say, French, this works reasonably well. A typical move in French is the switching of adjectives and nouns, such as when *affaires extérieur* becomes *external affairs*. The improvement in language model score for *external affairs* over *affairs external* is much higher than the reordering cost involved in the movement.

Given that reordering is mostly driven by the language model, and grudgingly allowed by the reordering model, one has to consider the limitations of the language model used in phrase-based statistical machine translation. It is typically based on trigrams, so only a window of three words is considered in making decisions on what is good English. This window is too small for making adequate judgments about overall grammaticalness of the sentence.

Given the weaknesses of the reordering model, it may not come as a surprise that limiting reordering to **monotone translation** is not very harmful. Allowing no reordering at all has other benefits: the search problem for finding the optimal translation according to the model is reduced in complexity from exponential to polynomial, making search algorithms much faster.

monotone translation

Allowing **limited reordering**, however, yields better translation results than allowing no reordering at all. If we permit moves within a window of a few words, we allow the local reordering required when translating Arabic–English (subject–verb, adjective–noun) or French–English (adjective–noun). Since this is also something that the language model can handle, it often represents the best we can do with reordering. Larger reordering windows or completely unrestricted reordering often leads to worse translations.

limited reordering

5.4.2 Lexicalized Reordering

The reordering model that we proposed for phrase-based statistical machine translation is only conditioned on movement distance and nothing else. However, as we have observed, some phrases are reordered more frequently than others. For instance, a French adjective like *extérieur* typically gets switched with the preceding noun, when translated into English.

Hence, we want to consider a **lexicalized reordering model** that conditions reordering on the actual phrases. One concern, of course, is the problem of sparse data. A particular phrase pair may occur only a few times in the training data, making it hard to estimate reliable probability distributions from these statistics.

lexicalized reordering model

Therefore, in the lexicalized reordering model we present here, we consider only three reordering types: (m) monotone order; (s) swap with previous phrase; and (d) discontinuous. Figure 5.8 illustrates these three different types of **orientation** of a phrase.

orientation

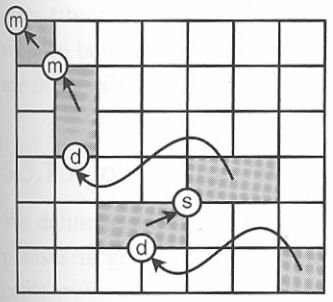


Figure 5.8 Three different orientations of phrases in the lexicalized reordering model: (m) monotone; (s) swap; (d) discontinuous.

To put it more formally, we want to introduce a reordering model p_o that predicts an orientation type m,s,d given the phrase pair currently used in translation:

$$\begin{aligned} &\text{orientation} \in \{\text{m, s, d}\} \\ &p_o(\text{orientation}|\bar{f}, \bar{e}) \end{aligned} \tag{5.10}$$

How can we learn such a probability distribution from the data? Again, we go back to the word alignment that was the basis for our phrase table. When we extract each phrase pair, we also extract its orientation type in that specific occurrence.

See Figure 5.9 for an illustration. Looking at the word alignment matrix, we note for each extracted phrase pair its corresponding orientation type (rows relate to English output words, columns to foreign input words). The orientation type can be detected if we check for word alignment points to the top left or to the top right of the extracted phrase pair. An alignment point to the top left signifies that the preceding English word is aligned to the preceding foreign word. An alignment point to the top right indicates that the preceding English word is aligned to the following foreign word.

The orientation type is detected as follows:

- **monotone**: if a word alignment point to the top left exists, we have evidence for monotone orientation;
- **swap**: if a word alignment point to the top right exists, we have evidence for a swap with the previous phrase;
- **discontinuous**: if no word alignment point exists to top left or to the top right, we have neither monotone order nor a swap, and hence evidence for discontinuous orientation.

We count how often each extracted phrase pair is found with each of the three orientation types. The probability distribution p_o is then estimated based on these counts using the maximum likelihood principle:

$$p_o(\text{orientation}|\bar{f}, \bar{e}) = \frac{\text{count}(\text{orientation}, \bar{e}, \bar{f})}{\sum_o \text{count}(o, \bar{e}, \bar{f})} \tag{5.11}$$

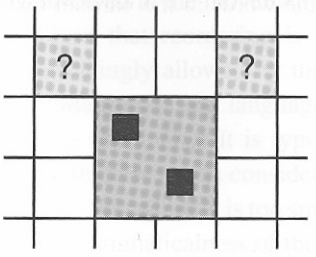


Figure 5.9 Evidence for different orientation types: Does a word alignment point exist to the top left or top right in the word alignment matrix?

Given the sparse statistics of the orientation types, we may want to smooth the counts with the unconditioned maximum-likelihood probability distribution with some factor σ:

$$p_o(\text{orientation}) = \frac{\sum_{\bar{f}} \sum_{\bar{e}} \text{count}(\text{orientation}, \bar{e}, \bar{f})}{\sum_o \sum_{\bar{f}} \sum_{\bar{e}} \text{count}(o, \bar{e}, \bar{f})} \quad (5.12)$$

$$p_o(\text{orientation}|\bar{f}, \bar{e}) = \frac{\sigma\, p(\text{orientation}) + \text{count}(\text{orientation}, \bar{e}, \bar{f})}{\sigma + \sum_o \text{count}(o, \bar{e}, \bar{f})} \quad (5.13)$$

There are a number of variations on this lexicalized reordering model based on orientation types:

- Certain phrases may not only flag, if they themselves are moved out of order, but also if subsequent phrases are reordered. A lexicalized reordering model for this decision could be learnt in addition, using the same method.
- Due to sparse data concerns, we may want to condition the probability distribution only on the foreign phrase (or the English phrase).
- To further reduce the complexity of the model, we might merge the orientation types swap and discontinuous, leaving a binary decision about **monotonicity** of the phrase order.

monotonicity

These variations have been shown to be occasionally beneficial for certain training corpus sizes and language pairs.

5.5 EM Training of Phrase-Based Models

We described in Section 5.2 a method for creating a phrase translation table from a word-aligned parallel corpus. The phrase alignment is done in two steps. First a word alignment is established using expectation maximization (EM) training. Then, we extract phrases that are consistent with the word alignment.

But why the detour over word alignments? Is it not possible to directly align phrases in a sentence pair? In this section, we will discuss a method that does just that.

Recall the intuition behind EM training for word-based models. We laid out a generative model that explains the data but has hidden parameters (the word alignment) that are not directly observable. Here, we want to build a phrase model. As before, the words in the parallel text are observable, but not the phrasal alignment between them.

5.5.1 A Joint Model for Phrasal Alignment

We define a generative model in which a pair of foreign and English phrases is generated by a **concept**. Formally, first a number of concepts $c = c_1 ... c_n$ are generated. Then each concept c_i generates a

concept

joint model

phrase pair $(\bar{e}_i, \bar{f}_i)$ with probability $p(\bar{e}_i, \bar{f}_i | c_i)$. Since both foreign and English phrases are generated by the model, this model is called a **joint model**. Each phrase is placed at a specific position *pos* in the sentence. The phrase pairs form the sentence pair $\mathbf{f} = \bar{f}_1 ... \bar{f}_n, \mathbf{e} = \bar{e}_1 ... \bar{e}_n$, with an English output order defined by *pos*. Hence, the combined probability of the sentence pair **e**,**f** generated by this generative process is

$$p(\mathbf{e}, \mathbf{f}) = \prod_{i=1}^{n} p(\bar{e}_i, \bar{f}_i | c_i)\ d(\text{pos}(e_i) | \text{pos}(e_{i-1})) \tag{5.14}$$

There are many ways to define the distortion (reordering) probability d. We may choose a fixed distance-based reordering model as in the standard model. Note that we can also learn the probability distribution properly from the parallel corpus, since we have a generative model that explains the data.

Figure 5.10 gives an example sentence pair. Five concepts $c_1, ..., c_5$ generate the sentence pair (*of course john has fun with the game*, *natuerlich hat john spass am spiel*). The first concept generates (*of course*, *natuerlich*), the second concept generates (*has*, *hat*), and so on. Note that the concepts c_2 and c_3 create phrases that are in different orders in German and English.

Although the notion of *concept* suggests that meaning is attached to these originators of phrase pairs, we actually only use one universal concept that generates all phrase pairs. Hence, the generating probability is simply $t(\bar{e}, \bar{f}) = p(\bar{e}, \bar{f} | \bar{c})$.

5.5.2 Complexity of the Alignment Space

Before we turn to training the model, first some comments on the complexity of the space of possible alignments. If we limit ourselves to contiguous phrases, a phrase may start at any position and end at any position thereafter. This means that from a sentence with length n, we can identify $O(n^2)$ phrases.

The complexity of possible phrases in the English sentence is $O(n^2)$, and the complexity of possible phrases in the foreign sentence is $O(n^2)$ as well. Without any prior knowledge, any phrase in the English

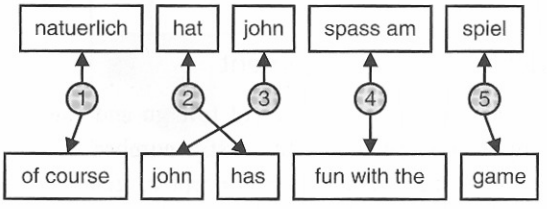


Figure 5.10 Example of a joint model that directly aligns foreign and English phrases: Five concepts generate the sentence pair.

sentence may be mapped to any phrase in the foreign sentence. So, a concept may generate any of $O(n^4)$ phrase pairs.

How many different ways are there to generate a sentence pair of length n each? First of all, m concepts have to be chosen with $1 \leq m \leq n$. If we generate phrases in the sequence of, say, English words, we still have $O(n^3)$ choices when aligning a phrase pair. Since we have to make m such choices, we end up with a complexity of $O(m^{n^3})$ for the space of possible phrasal alignments for a sentence pair, clearly a space that is too large to search exhaustively in a straightforward way.

Another problem we need to consider is the enormous inventory of possible phrases for each side of the parallel corpus, and the resulting size of the bilingual phrase translation table. During training, we have to consider not only the $O(n)$ different phrases of the highest-probability Viterbi alignment, but all $O(n^4)$ possible phrase pairs. Given modern training corpus sizes of maybe a million sentences, this number easily becomes too large to be stored in memory, or even on disk.

Contrast this with the phrase alignment method based on word alignments that we discussed before. The word alignment points effectively restrict the space of possible phrase alignments to a small fraction of the space considered for the joint model. In fact, this allows us to exhaustively collect all remaining possible phrase alignments as entries in the phrase translation table.

5.5.3 Training the Model

Setting aside the complexities of the alignment space for a moment, the basic method for expectation maximization (EM) training of the joint phrase model is very simple:

- Initialize: We begin with a uniform joint model, meaning all $p(\bar{e},\bar{f})$ phrase pair probabilities are the same.
- Expectation step: We use the joint model to estimate the likelihood of all possible phrase alignments for all sentence pairs. This allows us to collect counts for phrase pairs $(\bar{e},\bar{f})$, weighted with the probability of the alignment in which they occur.
- Maximization step: Given the counts, we can update the estimate of the probability of the joint model $p(\bar{e},\bar{f})$ by maximum likelihood estimation.

Given the complexity of the alignment space, we need to take a few short-cuts to be able to train the model. First, we want to reduce the number of phrase pairs that we consider for inclusion in the phrase translation table. Secondly, we need to limit the space of possible alignments for count collection (expectation step).

To limit the phrase inventory, we consider only phrases (on each side) that occur at least a few times (say, five times). An exception is made for phrases consisting of only one word. We may also want to enforce a minimum number of co-occurrences of a phrase pair.

Let us now consider a method to make count collection for phrasal alignments for a particular sentence pair feasible (expectation step). You may recall the analogous discussion for the word-based IBM Models 3–5 (see Section 4.4.2 from page 100 onward). There, as here, we have to deal with an exponential space of possible alignments, which we cannot efficiently evaluate exhaustively for long sentences.

greedy search heuristic

Instead, we employ a **greedy search heuristic** that is efficient enough to find a high-probability word alignment in reasonable time, but is not guaranteed to find the optimal alignment. Count collection then proceeds to sample the space around the best alignment found by the search heuristic.

The greedy search for the best alignment works as follows. First, we greedily create an initial alignment, by using the highest probability $t(\bar{e},\bar{f})$ entries in the phrase translation table. Then, we hill-climb towards the highest probability alignment by (a) breaking and merging concepts, (b) moving words across concepts, and (c) swapping words between concepts.

High-probability alignments for collecting counts may be sampled in the neighborhood of the highest probability alignment. Alternatively, all phrase alignments seen during the greedy hill-climbing process may be taken as the sample for collecting counts.

This training procedure makes the direct EM training of the joint phrase model feasible, but it is still a computationally expensive process, in terms of both time and space. The application of this training routine is currently limited to small corpora, and even for those, cluster computers are commonly used to perform the training. In terms of performance, results are generally no better than learning a phrase table using word alignments, the method we described earlier.

5.6 Summary

5.6.1 Core Concepts

This chapter introduced **phrase-based** statistical machine translation models, which are based on the translation of **phrases** instead of words as atomic units. We define a phrase as a contiguous multiword sequence, without any linguistic motivation. Phrases are mapped one-to-one based on a **phrase translation table**, and may be reordered.

The phrase translation table may be learnt based on a word alignment. All phrase pairs that are **consistent with the word alignment** are added to the phrase table. At the end of the chapter we presented an alternative method, which learns phrasal alignments directly from a parallel corpus. According to a **joint model**, each phrase pair is generated from a **concept**. This model is trained using the expectation maximization (EM) algorithm, in a way similar to the word-based IBM model presented in the previous chapter.

We initially presented a simple **distance-based reordering** model, and then suggested a **lexicalized reordering** model. The lexicalized reordering model predicts the **orientation** of a phrase: either **monotone**, **discontinuous**, or **swap**. One variant of the orientation model checks only for **monotonicity** of the phrases.

Different model components in the phrase model are combined in a **log-linear model**, in which each component is a factor which may be weighted. We extended the original translation model by introducing additional model components: **bidirectional translation** probabilities, **lexical weighting**, **word penalty** and **phrase penalty**.

5.6.2 Further Reading

Introduction – Modern statistical phrase-based models are rooted in work by Och and Weber [1998]; Och *et al.* [1999]; Och [2002]; and Och and Ney [2004] on alignment template models. These models defined phrases over word classes that were then instantiated with words. Translating with the use of phrases in a statistical framework was also proposed by Melamed [1997b]; Wang and Waibel [1998]; Venugopal *et al.* [2003]; and Watanabe *et al.* [2003]. Marcu [2001] proposes the use of phrases within word-based model decoding. The use of log-linear models was proposed by Och and Ney [2002]. Our description follows Koehn *et al.* [2003], which is similar to the model by Zens *et al.* [2002] and Zens and Ney [2004]. Tribble *et al.* [2003] suggest the use of overlapping phrases. Lopez and Resnik [2006] show the contribution of the different components of a phrase-based model.

Learning translation models – Several methods to extract phrases from a parallel corpus have been proposed. Most make use of word alignments [Tillmann, 2003; Zhang *et al.*, 2003; Zhao and Vogel, 2005; Zhang and Vogel, 2005a; Setiawan *et al.*, 2005]. One may restrict extraction of phrase pairs to the smallest phrases that cover the sentence [Mariño *et al.*, 2005]. Lambert and Banchs [2005] compare this restrictive method with the method described in this book and

propose some refinements. Phrase alignment may be done directly from sentence-aligned corpora using a probabilistic model [Shin *et al.*, 1996], pattern mining methods [Yamamoto *et al.*, 2003], or matrix factorization [Goutte *et al.*, 2004]. IBM Model 1 probabilities may be used to separate words aligned to each phrase from words outside it [Vogel, 2005], a method also used for splitting long sentences [Xu *et al.*, 2005]. Zhao *et al.* [2004b] use a measure based on the td-idf score from information retrieval to score phrase translations. Additional feature scores may also be used during the parameter tuning of the decoder to determine which phrase pairs should be discarded [Deng *et al.*, 2008]. Kim and Vogel [2007] use an iterative method that adds extracted phrases to the parallel corpus to bootstrap better alignments and extract better phrases. Turchi *et al.* [2008] give an overall analysis of the learning problem for phrase-based machine translation.

EM training of phrase models – The joint phrase model we described is taken from Marcu and Wong [2002], who also propose an improved model initialization over that presented here. The joint model may also be improved by constraining it with alignment points from the intersection of IBM model alignments [Birch *et al.*, 2006a,b] or by not strictly requiring a unique phrase alignment [Moore and Quirk, 2007b]. DeNero *et al.* [2006] point to some problems when using EM training with conditional probabilities. Cherry and Lin [2007] show that the ITG constraint helps the joint phrase model approach, partly by enabling a faster algorithm with fewer search errors. The phrase alignment problem is NP-complete [DeNero and Klein, 2008].

Refinements of phrase models – Usually, the segmentation of the source is not modeled, or only a phrase count feature is used, but adding a source phrase segmentation model may be beneficial [Blackwood *et al.*, 2008b]. Models may allow word insertion to account for spurious function words [Xu, 2005], or allow for words to be dropped by translating them into the empty phrase [Li *et al.*, 2008].

Context features in translation – Phrase translation may be informed by additional context features, for instance by applying methods used in word sense disambiguation [Carpuat *et al.*, 2006]. Such features may be integrated using a maximum entropy model [Bangalore *et al.*, 2006], using support vector machines [Giménez and Màrquez, 2007a], or by directly integrating more complex word sense disambiguation components, such as an ensemble of different machine learning methods [Carpuat and Wu, 2007a,b]. Ittycheriah and Roukos [2007] propose a maximum entropy model for phrase translation. Syntactic context dependencies may be added to phrase translations in the phrase-based approach, for instance verb–argument relationships

[Hwang and Sasaki, 2005], or the syntactic structure underlying each phrase translation [Sun *et al.*, 2007]. Gimpel and Smith [2008] add features around the context of a source phrase into a probabilistic back-off model.

Smoothing – Smoothing the phrase translation probability using discounting methods that are common in language modeling has been shown to improve performance [Foster *et al.*, 2006]. Continuous space models implemented as neural networks allow smoothing of phrase translation table probabilities [Schwenk *et al.*, 2007].

Paraphrasing – More robust translation models may be generated by paraphrasing phrase translation entries [Callison-Burch *et al.*, 2006a] or the parallel corpus [Nakov and Hearst, 2007]. Paraphrases may be extracted from a parallel corpus [Bannard and Callison-Burch, 2005]; for more accuracy the dependency structure may be exploited [Hwang *et al.*, 2008].

Use of dictionaries – Existing bilingual dictionaries may simply be added as additional parallel data to the training data. This can, however, miss the right context in which these words occur. Okuma *et al.* [2007] propose inserting phrases into the phrase tables that adapt existing entries with a very similar word to the dictionary word by replacing it with the dictionary word.

Pruning large translation models – Quirk and Menezes [2006] argue that extracting only minimal phrases, i.e. the smallest phrase pairs that map each entire sentence pair, does not hurt performance. This is also the basis of the n-gram translation model [Mariño *et al.*, 2006; Costa-jussà *et al.*, 2007], a variant of the phrase-based model. Discarding unlikely phrase pairs based on significance tests on their more-than-random occurrence reduces the phrase table drastically and may even yield increases in performance [Johnson *et al.*, 2007]. Wu and Wang [2007a] propose a method for filtering the noise in the phrase translation table based on a log likelihood ratio. Kutsumi *et al.* [2005] use a support vector machine for cleaning phrase tables. Such considerations may also be taken into account in a second-pass phrase extraction stage that does not extract bad phrase pairs [Zettlemoyer and Moore, 2007]. When faced with porting phrase-based models to small devices such as PDAs [Zhang and Vogel, 2007], the translation table has to be reduced to fit a fixed amount of memory. Eck *et al.* [2007a,b] prune the translation table based on how often a phrase pair was considered during decoding and how often it was used in the best translation.

Suffix arrays for storing translation models – With the increasing size of available parallel corpora and translation models, efficient

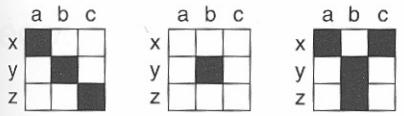
use of working memory becomes an issue, motivating the development of parallel infrastructures for training such as Google's MapReduce [Dyer *et al.*, 2008a]. Alternatively, the translation table may be represented in a suffix array as proposed for a searchable translation memory [Callison-Burch *et al.*, 2005a] and integrated into the decoder [Zhang and Vogel, 2005b]. Callison-Burch *et al.* [2005b] propose a suffix-tree structure to keep corpora in memory and extract phrase translations on the fly. The hierarchical phrase-based model (see Chapter 11) may also be stored in such a way [Lopez, 2007] and allows for much bigger models [Lopez, 2008b]. Suffix arrays may also be used to quickly learn phrase alignments from a parallel corpus without the use of a word alignment [McNamee and Mayfield, 2006]. Related to this is the idea of prefix data structures for the translation which allow quicker access and storage of the model on disk for on-demand retrieval of applicable translation options [Zens and Ney, 2007].

Lexicalized reordering – The lexicalized reordering model was first proposed by Tillmann [2004]; our description follows Koehn *et al.* [2005]. A similar model was proposed by Ohashi *et al.* [2005]. These models have been integrated in finite-state implementations [Kumar and Byrne, 2005]. Nagata *et al.* [2006] extend lexicalized reordering models with better generalization by conditioning on words instead of phrases and clustering words into classes. Similarly, Xiong *et al.* [2006] and Zens and Ney [2006a] use a maximum entropy classifier to learn better reordering probabilities. Features over the source syntax tree may be included in such reordering models [Xiong *et al.*, 2008b] as well as in the translation model [Xiong *et al.*, 2008a]. They may also predict reordering distance [Al-Onaizan and Papineni, 2006].

Phrase-based models and example-based translation – Phrase-based SMT is related to example-based machine translation (EBMT) [Somers, 1999]. Some recent systems blur the distinction between the two fields [Groves and Way, 2005; Paul *et al.*, 2005; Tinsley *et al.*, 2008]. Various combinations of methods from SMT and EBMT are explored by Groves and Way [2006]. Similar convergence takes place when combining statistical machine translation with translation memory, for instance by looking for similar sentences in the training data and replacing the mismatch with a translation chosen with statistical translation methods [Hewavitharana *et al.*, 2005]. Along these lines, phrase-based models may be improved by dynamically constructing translations for unknown phrases by using similar phrases that differ in a word or two and inserting lexical translations for the mismatched words [He *et al.*, 2008b]. Statistical machine translation models may be used to select the best translation from several example-based systems [Paul and Sumita, 2006].

5.6.3 Exercises

1. (⋆) Consider the following three-word alignment examples.



   For each, which and how many phrase pairs can be extracted? What do these examples suggest about the relationship between number of alignment points and number of extracted phrase pairs?
2. (⋆) Explain how the estimation of phrase translation probabilities can be efficiently handled without keeping all phrase pairs in memory, but keeping them in a big file on disk.
3. (⋆) The proposed lexicalized reordering model learns an orientation probability distribution for every phrase pair. What simplified alternatives would you suggest?
4. (⋆) If the phrase model were extended to allow phrases that have gaps on the input side only, what does this mean for the complexity of the phrase table?
5. (⋆⋆) Sketch out a pseudo-code implementation for EM training of the joint model.
6. (⋆⋆) Implement the phrase extraction algorithm and test it on a word-aligned parallel corpus.