Chapter 5
Phrase-Based Models

The currently best performing statistical machine translation systems are based on phrase-based models: models that translate small word sequences at a time. This chapter explains the basic principles of phrase-based models and how they are trained, and takes a more detailed look at extensions to the main components: the translation model and the reordering model. The next chapter will explain the algorithms that are used to translate sentences using these models.

5.1 Standard Model

First, we lay out the standard model for phrase-based statistical machine translation. While there are many variations, these can all be seen as extensions to this model.

5.1.1 Motivation for Phrase-Based Models

The previous chapter introduced models for machine translation that were based on the translation of words. But words may not be the best candidates for the smallest units for translation. Sometimes one word in a foreign language translates into two English words, or vice versa. Word-based models often break down in these cases.

Consider Figure 5.1, which illustrations how phrase-based models work. The German input sentence is first segmented into so-called **phrases** (any multiword units). Then, each phrase is translated into an English phrase. Finally, phrases may be reordered. In Figure 5.1, the six German words and eight English words are mapped as five phrase pairs.

phrase

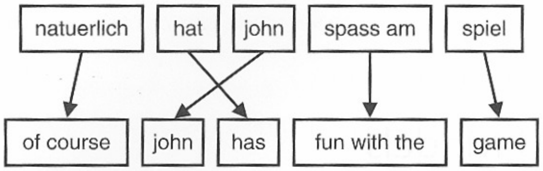


Figure 5.1 Phrase-based machine translation: The input is segmented into phrases (not necessarily linguistically motivated), translated one-to-one into phrases in English and possibly reordered.

The English phrases have to be reordered, so that the verb follows the subject.

phrase translation table

The German word *natuerlich* best translates into *of course*. To capture this, we would like to have a translation table that maps not words but phrases. A **phrase translation table** of English translations for the German *natuerlich* may look like the following:

Translation	Probability *p(e|f)*
of course	0.5
naturally	0.3
of course ,	0.15
, of course ,	0.05

It is important to point out that current phrase-based models are not rooted in any deep linguistic notion of the concept phrase. One of the phrases in Figure 5.1 is *fun with the*. This is an unusual grouping. Most syntactic theories would segment the sentence into the noun phrase *fun* and the prepositional phrase *with the game*.

However, learning the translation of *spass am* as *fun with the* is very useful. German and English prepositions do not match very well. But the context provides useful clues about how they have to be translated. The German *am* has many possible translations in English. Translating it as *with the* is rather unusual (more common is *on the* or *at the*), but in the context of following *spass* it is the dominant translation.

Let's recap. We have illustrated two benefits of translation based on phrases instead of words. For one, words may not be the best atomic units for translation, due to frequent one-to-many mappings (and vice versa). Secondly, translating word groups instead of single words helps to resolve translation ambiguities. There is a third benefit: if we have large training corpora, we can learn longer and longer useful phrases, sometimes even memorize the translation of entire sentences. Finally, the model is conceptually much simpler. We do away with the complex notions of fertility, insertion and deletion of the word-based model.

Intuitively, a model that does not allow the arbitrary adding and dropping of words makes more sense.

5.1.2 Mathematical Definition

Let us now define the phrase-based statistical machine translation model mathematically. First, we apply the Bayes rule to invert the translation direction and integrate a language model p_{LM}. Hence, the best English translation $\mathbf{e}_{\text{best}}$ for a foreign input sentence $\mathbf{f}$ is defined as

$$\begin{aligned}\mathbf{e}_{\text{best}} &= \text{argmax}_{\mathbf{e}}\ p(\mathbf{e}|\mathbf{f}) \\ &= \text{argmax}_{\mathbf{e}}\ p(\mathbf{f}|\mathbf{e})\ p_{\text{LM}}(\mathbf{e}) \qquad (5.1)\end{aligned}$$

This is exactly the same reformulation that we have already seen for word-based models (see Equation 4.23). For the phrase-based model, we decompose $p(\mathbf{f}|\mathbf{e})$ further into

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\ d(\text{start}_i - \text{end}_{i-1} - 1) \qquad (5.2)$$

The foreign sentence $\mathbf{f}$ is broken up into I phrases $\bar{f}_i$. Note that this process of segmentation is not modeled explicitly. This means that any segmentation is equally likely.

Each foreign phrase $\bar{f}_i$ is translated into an English phrase $\bar{e}_i$. Since we mathematically inverted the translation direction in the noisy channel, the phrase translation probability $\phi(\bar{f}_i|\bar{e}_i)$ is modelled as a translation from English to foreign.

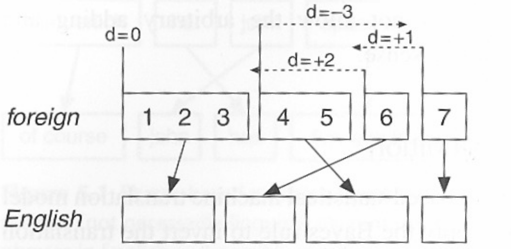
Reordering is handled by a **distance-based reordering model**. We consider reordering relative to the previous phrase. We define start_i as the position of the first word of the foreign input phrase that translates to the ith English phrase, and end_i as the position of the last word of that foreign phrase. Reordering distance is computed as $\text{start}_i - \text{end}_{i-1} - 1$.

distance-based reordering model

The reordering distance is the number of words skipped (either forward or backward) when taking foreign words out of sequence. If two phrases are translated in sequence, then $\text{start}_i = \text{end}_{i-1} + 1$; i.e., the position of the first word of phrase i is the same as the the position of the last word of the previous phrase plus one. In this case, a reordering cost of $d(0)$ is applied. See Figure 5.2 for an example.

What is the probability of d? Instead of estimating reordering probabilities from data, we apply an exponentially decaying cost function $d(x) = \alpha^{|x|}$ with an appropriate value for the parameter $\alpha \in [0, 1]$ so that d is a proper probability distribution.[1] This formula simply means

[1] Actually, we do not worry too much about turning d into a proper probability distribution, because we weight model components according to their importance in a log-linear model. We will describe this in Section 5.3.1 on page 136.



Phrase	Translates	Movement	Distance
1	1–3	start at beginning	0
2	6	skip over 4–5	+2
3	4–5	move back over 4–6	−3
4	7	skip over 6	+1

Figure 5.2 Distance-based reordering: Reordering distance is measured on the foreign input side. In the illustration each foreign phrase is annotated with a dashed arrow indicating the extent of reordering. For instance the 2nd English phrase translates the foreign word 6, skipping over the words 4–5, a distance of +2.

that movements of phrases over large distances are more expensive than shorter movements or no movement at all.

Note that this reordering model is very similar to the one in word-based models. We could even learn reordering probabilities from the data, but this is not typically done in phrase-based models.

What we have just described is a simple phrase-based statistical machine translation model. Only the phrase translation table is learnt from data, reordering is handled by a predefined model. We will describe one method to learn a phrase translation table in the next section and then discuss some extensions to the standard model, both to the translation model and to the reordering model.

5.2 Learning a Phrase Translation Table

Clearly, the power of phrase-based translation rests on a good phrase translation table. There are many ways to acquire such a table. We will present here one method in detail. First, we create a word alignment between each sentence pair of the parallel corpus, and then **extract phrase pairs** that are consistent with this word alignment.

phrase extraction

5.2.1 Extracting Phrases from a Word Alignment

Consider the word alignment in Figure 5.3, which should be familiar from the previous chapter. Given this word alignment we would like to extract phrase pairs that are consistent with it, for example matching the English phrase *assumes that* with the German phrase *geht davon aus, dass*.

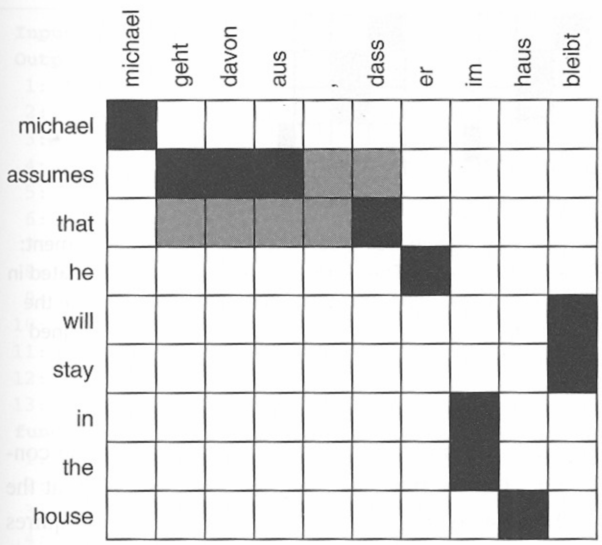


Figure 5.3 Extracting a phrase from a word alignment: The English phrase *assumes that* and the German phrase *geht davon aus, dass* are aligned, because their words are aligned to each other.

If we have to translate a German sentence that contains the phrase *geht davon aus, dass* then we can use the evidence of this phrasal alignment to translate the phrase as *assumes that*. Useful phrases for translation may be shorter or longer than this example. Shorter phrases occur more frequently, so they will more often be applicable to previously unseen sentences. Longer phrases capture more local context and help us to translate larger chunks of text at one time, maybe even occasionally an entire sentence.

Hence, when extracting phrase pairs, we want to collect both short and long phrases, since all of them are useful.

5.2.2 Definition of Consistency

Coming back to the example in Figure 5.3, we collected each phrase pair from the sentence pair using the word alignment because its words match up consistently. Let us put the definition of **consistent with a word alignment** on a more formal footing.

consistent with a word alignment

We call a phrase pair $(\bar{f}, \bar{e})$ consistent with an alignment A, if all words $f_1, ..., f_n$ in $\bar{f}$ that have alignment points in A have these with words $e_1, ..., e_n$ in $\bar{e}$ and vice versa:

$$
\begin{aligned}
&(\bar{e}, \bar{f}) \text{ consistent with } A \Leftrightarrow \\
&\qquad \forall e_i \in \bar{e} : (e_i, f_j) \in A \Rightarrow f_j \in \bar{f} \\
&\qquad \text{AND } \forall f_j \in \bar{f} : (e_i, f_j) \in A \Rightarrow e_i \in \bar{e} \\
&\qquad \text{AND } \exists e_i \in \bar{e}, f_j \in \bar{f} : (e_i, f_j) \in A
\end{aligned}
\tag{5.3}
$$

Figure 5.4 illustrates what kind of phrase pairs this definition includes and excludes. Note especially the case of unaligned words.

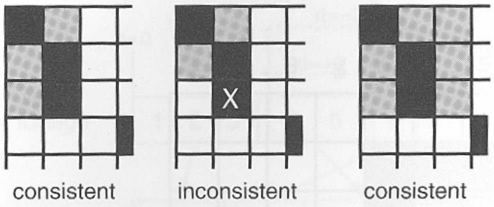


Figure 5.4 Definition of phrase pairs being consistent with a word alignment: All words have to align to each other. This is true in the first example, violated in the second example (one alignment point in the second column is outside the phrase pair), and true in the third example (note that it includes an unaligned word on the right).

Since they do not have alignment points, they cannot violate the condition for consistency. Hence, they may occur within and even at the edges of a phrase. The last condition in the definition, however, requires that the phrase pair includes at least one alignment point.

5.2.3 Phrase Extraction Algorithm

Given the definition of consistency, we now want to devise an algorithm that extracts all consistent phrase pairs from a word-aligned sentence pair.

Such an algorithm is sketched out in Figure 5.5. The idea is to loop over all possible English phrases and find the minimal foreign phrase that matches each of them. Matching is done by identifying all alignment points for the English phrase and finding the shortest foreign phrase that includes all the foreign counterparts for the English words.

The following has to be taken into account:

- If the English phrase contains only unaligned words, we do not want to match it against the foreign sentence.
- If the matched minimal foreign phrase has additional alignment points outside the English phrase, we cannot extract this phrase pair. In fact, no phrase pair can be extracted for this English phrase.
- Other foreign phrases than the minimally matched foreign phrase may be consistent with the English phrase. If the foreign phrase borders unaligned words, then it is extended to these words, and the extended phrase is also added as a translation of the English phrase.

One way to look at the role of alignment points in extracting phrases is that they act as constraints for which phrase pairs can be extracted. The fewer alignment points there are, the more phrase pairs can be extracted (this observation is not valid in the extreme: with no alignment points at all, no phrase pairs can be extracted).

```
Input: word alignment $A$ for sentence pair $(\mathbf{e}, \mathbf{f})$
Output: set of phrase pairs $BP$
for $e_{\text{start}} = 1 \ldots \text{length}(\mathbf{e})$ do
  for $e_{\text{end}} = e_{\text{start}} \ldots \text{length}(\mathbf{e})$ do
    // find the minimally matching foreign phrase
    $(f_{\text{start}}, f_{\text{end}}) = (\text{length}(\mathbf{f}), 0)$
    for all $(e,f) \in A$ do
      if $e_{\text{start}} \leq e \leq e_{\text{end}}$ then
        $f_{\text{start}} = \min(f, f_{\text{start}})$
        $f_{\text{end}} = \max(f, f_{\text{end}})$
      end if
    end for
    add extract($f_{\text{start}}, f_{\text{end}}, e_{\text{start}}, e_{\text{end}}$) to set $BP$
  end for
end for
function extract($f_{\text{start}}, f_{\text{end}}, e_{\text{start}}, e_{\text{end}}$)
return {} if $f_{\text{end}} == 0$ // check if at least one alignment point
// check if alignment points violate consistency
for all $(e,f) \in A$ do
  return {} if $e < e_{\text{start}}$ or $e > e_{\text{end}}$
end for
// add pharse pairs (incl. additional unaligned f)
E = {}
$f_s = f_{\text{start}}$
repeat
  $f_e = f_{\text{end}}$
  repeat
    add phrase pair ($e_{\text{start}} \ldots e_{\text{end}}$, $f_s \ldots f_e$) to set $E$
    $f_e{+}{+}$
  until $f_e$ aligned
  $f_s{-}{-}$
until $f_s$ aligned
return $E$
```

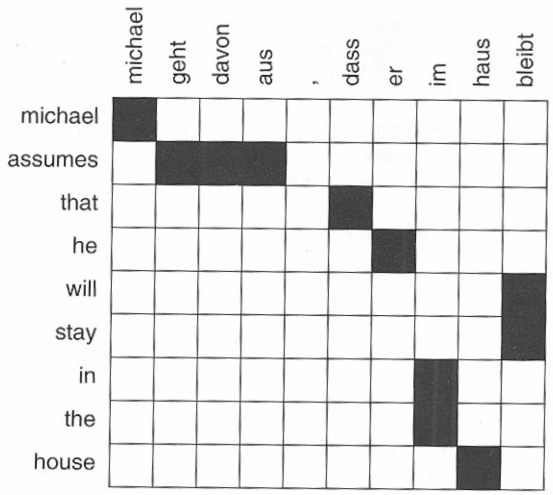
Figure 5.5 Phrase extraction algorithm: For each English phrase $e_{\text{start}} \ldots e_{\text{end}}$, the minimal phrase of aligned foreign words is identified $f_{\text{start}} \ldots f_{\text{end}}$. Words in the foreign phrase are not allowed to be aligned with English words outside the English phrase. This pair of phrases is added, along with additional phrase pairs that include additional unaligned foreign words at the edge of the foreign phrase.

5.2.4 Example

Let us turn back to our example sentence pair (from Figure 5.3). What phrase pairs are consistent with the word alignment, and hence will be extracted by our algorithm? Figure 5.6 displays the complete list.

It is possible that for some English phrases, we are not able to extract matching German phrases. This happens, for instance, when multiple English words are aligned to one German word: *in the* are both aligned to the German *im*, so that no individual match for either *in* or *the* can be extracted.

Figure 5.6 Extracted phrase pairs from the word alignment in Figure 5.3: For some English phrases, multiple mappings are extracted (e.g., *that* translates to *dass* with and without preceding comma); for some English phrases, no mappings can be found (e.g., *the* or *he will*).



michael – michael
michael assumes – michael geht davon aus ; michael geht davon aus ,
michael assumes that – michael geht davon aus , dass
michael assumes that he – michael geht davon aus , dass er
michael assumes that he will stay in the house
– michael geht davon aus , dass er im haus bleibt
assumes – geht davon aus ; geht davon aus ,
assumes that – geht davon aus , dass
assumes that he – geht davon aus , dass er
assumes that he will stay in the house
– geht davon aus , dass er im haus bleibt
that – dass ; , dass
that he – dass er ; , dass er
that he will stay in the house
– dass er im haus bleibt ; , dass er im haus bleibt
he – er
he will stay in the house – er im haus bleibt
will stay – bleibt
will stay in the house – im haus bleibt
in the – im
in the house – im haus
house – haus

This also happens when the English words align with German words that enclose other German words that align back to English words that are not in the original phrase. See the example of *he will stay*, which aligns to *er ... bleibt*, words that enclose *im haus*, which aligns back to *in the house*. Here, it is not possible to match *he will stay* to any German phrase, since the only matching German phrase has a gap.

Unaligned words may lead to multiple matches for an English phrase: for instance, *that* matches to *dass* with and without the preceding unaligned comma on the German side.

Note some numbers for this example: there are 9 English words and 10 German words, matched by 11 alignment points. There are 36 distinct contiguous English phrases and 45 distinct contiguous German phrases. 24 phrase pairs are extracted.

Obviously, allowing phrase pairs of any length leads to a huge number of extracted phrase pairs. In well-behaved alignments without reordering, the number of extracted phrases is roughly quadratic in the number of words. However, most long phrases observed in the training data will never occur in the test data. Hence, to reduce the number of extracted phrases and keep the phrase translation table manageable, we may want to enforce a maximum phrase length.

Another reason there are a huge number of extracted phrases is unaligned words. Observe the effect of the unaligned comma on the German side. If it were aligned to *that*, five fewer phrase pairs would be extractable. However, while handling a large number of extracted phrases may cause computational problems, it is less clear whether this hinders our ultimate purpose: improving the quality of the output of our machine translation system.

5.2.5 Estimating Phrase Translation Probabilities

So far, we have only discussed how to collect a set of phrase pairs. More is needed to turn this set into a probabilistic phrase translation table.

It is worth noting that what unfolds here is different from the generative modeling of the IBM models, presented in the previous chapter. Previously, we had a mathematical model that explained, in a generative story, how words in the input sentence are translated into words in the output sentence. This story gave different probabilities for different alignments between input and output sentences, and counts for word translation (and other model components) were based on the relative weight of these alignments.

In contrast to this, here we do not choose among different phrase alignments. Quite purposely, we do not make a choice between, for instance, a more fine-grained alignment with many small phrases or a coarser alignment with a few large phrases. Phrases of any length may come in handy, and we do not want to eliminate any of them.

These practical considerations lead us to a different estimation technique for the conditional probability distributions of the phrase translation table. For each sentence pair, we extract a number of phrase pairs. Then, we count in how many sentence pairs a particular phrase pair is extracted and store this number in count$(\bar{e},\bar{f})$. Finally, the translation probability $\phi(\bar{f}|\bar{e})$ is estimated by the relative frequency:

$$\phi(\bar{f}|\bar{e}) = \frac{\text{count}(\bar{e},\bar{f})}{\sum_{\bar{f}_i} \text{count}(\bar{e},\bar{f}_i)} \tag{5.4}$$

We may want to take into consideration the case when one phrase is matched to multiple phrases in a particular sentence pair, which frequently occurs when there are unaligned words. To reflect the degree of uncertainty, we could assign, for each of the matches, fractional counts that add up to one.

Size of the phrase table

For large parallel corpora of millions of sentences, the extracted phrase translation tables easily require several gigabytes of memory. This may be too much to fit into the working memory of a machine. This causes problems for estimating phrase translation probabilities and later the use of these tables to translate new sentences.

For the estimation of the phrase translation probabilities, not all phrase pairs have to be loaded into memory. It is possible to efficiently estimate the probability distribution by storing and sorting the extracted phrases on disk. Similarly, when using the translation table for the translation of a single sentence, only a small fraction of it is needed and may be loaded on demand.

5.3 Extensions to the Translation Model

So far in this chapter, we have described the standard model for phrase-based statistical machine translation. Even this relatively simple version achieves generally better translation quality than the word-based statistical IBM models. In the rest of this chapter, we will extend the model, achieving further improvement of translation performance.

5.3.1 Log-Linear Models

The standard model described so far consists of three factors:

- the phrase translation table $\phi(\bar{f}|\bar{e})$;
- the reordering model d;
- the language model $p_{\text{LM}}(e)$.

These three model components are multiplied together to form our **phrase-based statistical machine translation model**:

phrase-based model

$$e_{\text{best}} = \text{argmax}_e \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\; d(\text{start}_i - \text{end}_{i-1} - 1) \prod_{i=1}^{|\mathbf{e}|} p_{\text{LM}}(e_i|e_1 ... e_{i-1}) \tag{5.5}$$

Another way to describe this setup is that there are three components that contribute to producing the best possible translation, by ensuring that

- the foreign phrases match the English words (ϕ);
- phrases are reordered appropriately (d);
- the output is fluent English (p_{LM}).

When we use our system, we may observe that the words between input and output match up pretty well, but that the output is not very good English. Hence, we are inclined to give the language model more weight. Formally, we can do this by introducing **weights** $\lambda_\phi, \lambda_d, \lambda_{\text{LM}}$ that let us scale the contributions of each of the three components:

weighting of components

$$e_{\text{best}} = \text{argmax}_e \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)^{\lambda_\phi}\ d(\text{start}_i - \text{end}_{i-1} - 1)^{\lambda_d} \prod_{i=1}^{|\mathbf{e}|} p_{\text{LM}}(e_i|e_1 ... e_{i-1})^{\lambda_{\text{LM}}} \tag{5.6}$$

What have we done here? Our original justification for decomposing the model into a translation model and a language model was the noisy-channel model. We applied the Bayes rule, which is a mathematically correct transformation. However, we followed that up with a number of independence assumptions that are not strictly correct, but are necessary to decompose the model further into probability distributions for which we have sufficient statistics.

The assumption behind the translation model that the translation of a phrase does not depend on surrounding phrases is such a necessary but inaccurate assumption. Similarly, the trigram language model assumption states that the probability of an English word depends only on a window of two previous words. It is not hard to come up with counterexamples for either of these assumption.

By adding weights, we are guided more by practical concerns than by mathematical rigor. However, we do come up with a model structure that is well known in the machine learning community: a **log-linear model**. Log-linear models have the following form:

log-linear model

$$p(x) = \exp \sum_{i=1}^{n} \lambda_i h_i(x) \tag{5.7}$$

Equation (5.6) fits this form with

- number of feature function $n = 3$;
- random variable $x = (e, f, \text{start}, \text{end})$;
- feature function $h_1 = \log \phi$;
- feature function $h_2 = \log d$;
- feature function $h_3 = \log p_{\text{LM}}$.

To make this more apparent, here is a reformulation of Equation (5.6):

$$p(e,a|f) = \exp\left[\lambda_\phi \sum_{i=1}^{I} \log \phi(\bar{f}_i|\bar{e}_i) + \lambda_d \sum_{i=1}^{I} \log d(a_i - b_{i-1} - 1) + \lambda_{\text{LM}} \sum_{i=1}^{|\mathbf{e}|} \log p_{\text{LM}}(e_i|e_1 ... e_{i-1})\right] \tag{5.8}$$

Log-linear models are widely used in the machine learning community. For instance, naive Bayes, maximum entropy, and perceptron learning methods are all based on log-linear models.

In this framework, we view each data point (here: a sentence translation) as a vector of features, and the model as a set of corresponding feature functions. The feature functions are trained separately, and combined assuming that they are independent of each other.

We already gave one reason for moving our model structure towards log-linear models: the weighting of the different model components may lead to improvement in translation quality. Another motivation is that this structure allows us naturally to include additional model components in the form of feature functions. We will do exactly this in the remainder of this chapter.

5.3.2 Bidirectional Translation Probabilities

The Bayes rule led us to invert the conditioning of translation probabilities: $p(\mathbf{e}|\mathbf{f}) = p(\mathbf{e})\ p(\mathbf{f}|\mathbf{e})\ p(\mathbf{f})^{-1}$. However, we may have some second thoughts about this in light of the phrase-based model we are now considering.

It may be that in the training data an unusual foreign phrase $\bar{f}$ exists that is mistakenly mapped to a common English phrase $\bar{e}$. In this case $\phi(\bar{f}|\bar{e})$ is very high, maybe even 1. If we encounter the phrase $\bar{f}$ again in the test data, this erroneous phrase translation will almost certainly be used to produce the highest probability translation: the translation model likes it – high $p(\bar{f}|\bar{e})$ – and the language model likes it as well, since $\bar{e}$ is a common English phrase.

In this case it would be better to use the conditioning of phrase translation probabilities in the actual translation direction, i.e., $\phi(\bar{e}|\bar{f})$. Having moved beyond the noisy-channel model, we may very well use the direct translation probabilities. It is even possible to use both **translation directions**, $\phi(\bar{e}|\bar{f})$ and $\phi(\bar{f}|\bar{e})$, as feature functions.

bidirectional translation

In practice, a model using both translation directions, with the proper weight setting, often outperforms a model that uses only the Bayes-motivated inverse translation direction, or only the direct translation direction.

5.3.3 Lexical Weighting

Some infrequent phrase pairs may cause problems, especially if they are collected from noisy data. If both of the phrases $\bar{e}, \bar{f}$ only occur once then $\phi(\bar{e}|\bar{f}) = \phi(\bar{f}|\bar{e}) = 1$. This often overestimates how reliable rare phrase pairs are.

How can we judge if a rare phrase pair is reliable? If we decompose it into its word translations, we can check how well they match up. This is called **lexical weighting**; it is basically a smoothing method. We back off to probability distributions (lexical translation), for which we have richer statistics and hence more reliable probability estimates.

lexical weighting

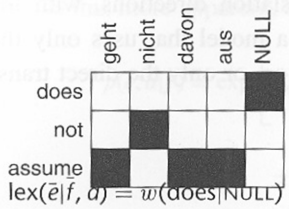
Many different lexical weighting methods have been proposed in the literature. Most of them are inspired by the word-based IBM models. Even using the relatively simple IBM Model 1 has been shown to be effective.

Let us describe one such weighting method. Recall that we extracted phrase pairs from a word alignment. Consequently, for each phrase pair, we also have the alignment between the words in the phrases available to us. Based on this alignment, we can compute the lexical translation probability of a phrase $\bar{e}$ given the phrase $\bar{f}$ by, for instance:

$$\text{lex}(\bar{e}|\bar{f}, a) = \prod_{i=1}^{\text{length}(\bar{e})} \frac{1}{|\{j|(i,j) \in a\}|} \sum_{\forall (i,j) \in a} w(e_i|f_j) \tag{5.9}$$

In this lexical weighting scheme, each of the English words e_i is generated by aligned foreign words f_j with the word translation probability $w(e_i|f_j)$. If an English word is aligned to multiple foreign words, the average of the corresponding word translation probabilities is taken. If an English word is not aligned to any foreign word, we say it is aligned to the NULL word, which is also factored in as a word translation probability.

In Figure 5.7 the English phrase *does not assume* is paired with the German *geht nicht davon aus*. The lexical weight for this phrase pair is the product of three factors, one for each English word. The English word *not* is aligned to *nicht*, so the factor is $w(\text{not}|\text{nicht})$. The English word *does* is not aligned to any foreign word, so the factor is $w(\text{does}|\text{NULL})$. The English word *assume* is aligned to three German words *geht davon aus*, so the factor is the average of the three corresponding word translation probabilities.



$$\text{lex}(\bar{e}|\bar{f}, a) = w(\text{does}|\text{NULL}) \times$$
$$w(\text{not}|\text{nicht}) \times$$
$$\frac{1}{3}(w(\text{assume}|\text{geht}) + w(\text{assume}|\text{davon}) + w(\text{assume}|\text{aus}))$$

Figure 5.7 Lexical weight p_w of a phrase pair $(\bar{e}, \bar{f})$ given an alignment a and a lexical translation probability distribution w: Each English word has to be explained by foreign words using the distribution w. If aligned to multiple foreign words, the average is taken. If unaligned, $w(e_i|\text{NULL})$ is factored in.

The lexical translation probabilities $w(e_i|f_j)$ are estimated from the word-aligned corpus. Counts are taken, and relative frequency estimation yields the probability distribution. Again, unaligned words are taken to be aligned to NULL.

Finally, as we pointed out in the previous section, it may be useful to use both translation directions in the model: $\text{lex}(\bar{e}|\bar{f}, a)$ and $\text{lex}(\bar{f}|\bar{e}, a)$.

5.3.4 Word Penalty

So far, we have not explicitly modeled the output length in terms of number of words. Yet one component of the system prefers shorter translations: the language model, simply because fewer trigrams have to be scored.

word penalty

To guard against output that is too short (or too long), we introduce a **word penalty** that adds a factor ω for each produced word. If $\omega < 1$ we increase the scores of shorter translations. If $\omega > 1$ we prefer longer translations.

This parameter is very effective in tuning output length and often improves translation quality significantly.

5.3.5 Phrase Penalty

Before any phrase translations can be applied to a new input sentence, the sentence has to be segmented into foreign phrases. This segmentation is not explicit in the model that we have presented so far. In effect, all segmentations are equally likely, and only the chosen phrase translations with their translation, reordering, and language model scores determine indirectly the input sentence segmentation.

phrase penalty

What is better: longer phrases or shorter phrases? A simple way to bias towards fewer and hence longer phrases or towards more and hence shorter phrases is to introduce a factor ρ for each phrase translation, a **phrase penalty**. Analogous to the word penalty ω discussed above, if