

# Entropy-based Pruning for Phrase-based Machine Translation

Wang Ling, João Graça, Isabel Trancoso, Alan Black

L<sup>2</sup>F Spoken Systems Lab, INESC-ID, Lisboa, Portugal

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

{wang.ling, joao.graca, isabel.trancoso}@inesc-id.pt

awb@cs.cmu.edu

## Abstract

Phrase-based machine translation models have shown to yield better translations than Word-based models, since phrase pairs encode the contextual information that is needed for a more accurate translation. However, many phrase pairs do not encode any relevant context, which means that the translation event encoded in that phrase pair is led by smaller translation events that are independent from each other, and can be found on smaller phrase pairs, with little or no loss in translation accuracy. In this work, we propose a relative entropy model for translation models, that measures how likely a phrase pair encodes a translation event that is derivable using smaller translation events with similar probabilities. This model is then applied to phrase table pruning. Tests show that considerable amounts of phrase pairs can be excluded, without much impact on the translation quality. In fact, we show that better translations can be obtained using our pruned models, due to the compression of the search space during decoding.

## 1 Introduction

Phrase-based Machine Translation Models (Koehn et al., 2003) model  $n$ -to- $m$  translations of  $n$  source words to  $m$  target words, which are encoded in phrase pairs and stored in the translation model. This approach has an advantage over Word-based Translation Models (Brown et al., 1993), since translating multiple source words allows the context for each source word to be considered during trans-

lation. For instance, the translation of the English word “in” by itself to Portuguese is not obvious, since we do not have any context for the word. This word can be translated in the context of “in (the box)” to “dentro”, or in the context of “in (China)” as “na”. In fact, the lexical entry for “in” has more than 10 good translations in Portuguese. Consequently, the lexical translation entry for Word-based models splits the probabilistic mass between different translations, leaving the choice based on context to the language model. On the other hand, in Phrase-based Models, we would have a phrase pair  $p(\textit{in the box}, \textit{dentro da caixa})$  and  $p(\textit{in china}, \textit{na china})$ , where the words “in the box” and “in China” can be translated together to “dentro da caixa” and “na China”, which substantially reduces the ambiguity. In this case, both the translation and language models contribute to find the best translation based on the local context, which generally leads to better translations.

However, not all words add the same amount of contextual information. Using the same example for “in”, if we add the context “(hid the key) in”, it is still not possible to accurately identify the best translation for the word “in”. The phrase extraction algorithm (Ling et al., 2010) does not discriminate which phrases pairs encode contextual information, and extracts all phrase pairs with consistent alignments. Hence, phrases that add no contextual information, such as,  $p(\textit{hid the key in}, \textit{escondeu a chave na})$  and  $p(\textit{hid the key in}, \textit{escondeu a chave dentro})$  are extracted. This is undesirable because we are populating translation models with redundant phrase pairs, whose translations can be obtained using com-

binations of other phrases with the same probabilities, namely  $p(\text{hid the key, escondeu a chave})$ ,  $p(\text{in, dentro})$  and  $p(\text{in, na})$ . This is a problem that is also found in language modeling, where large amounts of redundant higher-order n-grams can make the model needlessly large. For backoff language models, multiple pruning strategies based on relative entropy have been proposed (Seymore and Rosenfeld, 1996) (Stolcke, 1998), where the objective is to prune n-grams in a way to minimize the relative entropy between the model before and after pruning.

While the concept of using relative entropy for pruning is not new and frequently used in backoff language models, there are no such models for machine translation. Thus, the main contribution of our work is to propose a relative entropy pruning model for translation models used in Phrase-based Machine Translation. It is shown that our pruning algorithm can eliminate phrase pairs with little or no impact in the predictions made in our translation model. In fact, by reducing the search space, less search errors are made during decoding, which leads to improvements in translation quality.

This paper is organized as follows. We describe and contrast the state of the art pruning algorithms in section 2. In section 3, we describe our relative-entropy model for machine translation. Afterwards, in section 4, we apply our model for pruning in Phrase-based Machine Translation systems. We perform experiments with our pruning algorithm based on phrase pair independence and analyse the results in section 5. Finally, we conclude in section 6.

## 2 Phrase Table Pruning

Phrase table pruning algorithms are important in translation, since they efficiently reduce the size of the translation model, without having a large negative impact in the translation quality. This is especially relevant in environments where memory constraints are imposed, such as translation systems for small devices like cellphones, and also when time constraints for the translation are defined, such as online Speech-to-Speech systems.

### 2.1 Significance Pruning

A relevant reference in phrase table pruning is the work of (Johnson and Martin, 2007), where it is shown that a significant portion of the phrase table can be discarded without a considerable negative impact on translation quality, or even positive one. This work computes the probability, named p-value, that the joint occurrence event of the source phrase  $s$  and target phrase  $t$  occurring in same sentence pair happens by chance, and are actually statistically independent. Phrase pairs that have a high p-value, are more likely to be spurious and more prone to be pruned. This work is followed in (Tomeh et al., 2009), where phrase pairs are treated discriminately based on their complexity. Significance-based pruning has also been successfully applied in language modeling in (Moore and Quirk, 2009).

Our work has a similar objective, but instead of trying to predict the independence between the source and target phrases in each phrase pair, we attempt to predict the independence between a phrase pair and other phrase pairs in the model.

### 2.2 Relevance Pruning

Another proposed approach (Matthias Eck and Waibel, 2007) consists at collecting usage statistics for phrase pairs. This algorithm decodes the training corpora and extracts the number of times each phrase pair is used in the 1-best translation hypothesis. Thus, phrase pairs that are rarely used during decoding are excluded first during pruning.

This method considers the relationship between phrase pairs in the model, since it tests whether the decoder is more prone to use some phrase pairs than others. However, it leads to some undesirable pruning choices. Let us consider a source phrase “the box in China” and 2 translation hypotheses, where the first hypothesis uses the phrase translation  $p(\text{the key in China, a chave na China})$  with probability 70%, and the second hypothesis uses two phrase translations  $p(\text{the key, a chave})$  and  $p(\text{in China, na China})$  with probability 65%. This approach will lean towards pruning the phrase pairs in the second hypothesis, since the decoder will use the first hypothesis. This is generally not desired, since the 2 smaller phrase pairs can be used to translate the same source sentence with a small probab-

ity loss (5%), even if the longer phrase is pruned. On the other hand, if the smaller phrases are pruned, the longer phrase can not be used to translate smaller chunks, such as “the key in Portugal”. This matter is aggravated due to the fact that the training corpora is used to decode, so longer phrase pairs will be used more frequently than when translating unseen sentences, which will make the model more biased into pruning shorter phrase pairs.

### 3 Relative Entropy Model For Phrase-based Translation Models

In this section, we shall define our entropy model for phrase pairs. We start by introducing some notation to distinguish different types of phrase pairs and show why some phrase pairs are more redundant than others. Afterwards, we illustrate our notion of relative entropy between phrase pairs. Then, we describe our entropy model, its computation and its application to phrase table pruning.

#### 3.1 Atomic and Composite Phrase Pairs

We discriminate between 2 types of phrase pairs: atomic phrase pairs and composite phrase pairs.

Atomic phrase pairs define the smallest translation units, such that given an atomic phrase pair that translates from  $s$  to  $t$ , the same translation cannot be obtained using any combination of other phrase pairs. Removing these phrase pairs reduces the range of translations that our model is capable of translating and also the possible translations.

Composite phrase pairs define translations of a given sequence of words that can also be obtained using atomic or other smaller composite phrase pairs. Each combination is called a derivation or translation hypothesis. Removing these phrase pairs does not change the amount of sentences that the model can translate, since all translations encoded in these phrases can still be translated using other phrases, but these will lead to different translation probabilities.

Considering table 1, we can see that atomic phrases encode one elementary translation event, while composite phrases encode joint events that are encoded in atomic phrase pairs. If we look at the source phrase “in”, there is a multitude of possible translations for this word in most target languages.

Taking Portuguese as the target language, the probability that “in” is translated to “em” is relatively low, since it can also be translated to “no”, “na”, “dentro”, “dentro de” and many others.

However, if we add another word such as “Portugal” forming “in Portugal”, it is more likely that “in” is translated to “em”. Thus, we define the joint event of “in” translating to “em” ( $A_1$ ) and “Portugal” to “Portugal” ( $B_1$ ), denoted as  $A_1 \cap B_1$ , in the phrase pair  $p(\text{in Portugal}, \text{em Portugal})$ . Without this phrase pair it is assumed that these are independent events with probability given by  $P(A_1)P(B_1)$ <sup>1</sup>, which would be 10%, leading to a 60% reduction. In this case, it would be more likely, that *in Portugal* is translated to *no Portugal* or *na Portugal*, which would be incorrect.

Some words, such as “John”, forming “John in”, do not influence the translations for the word “in”, since it can still be translated to “em”, “no”, “na”, “dentro” or “dentro de” depending on the word that follows. By definition, if the presence of phrase  $p(\text{John}, \text{John})$  does not influence the translation of  $p(\text{in}, \text{em})$  and viceversa, we can say that probability of the joint event  $P(A_1 \cap C_1)$  is equal to the product of the probabilities of the events  $P(A_1)P(C_1)$ .

If we were given a choice of pruning either the composite phrase pairs  $p(\text{John in}, \text{John em})$  or  $p(\text{in Portugal}, \text{em Portugal})$ , the obvious choice would be the former, since the probability of the event encoded in that phrase pair is composed by 2 independent events, in which case the decoder will inherently consider the hypothesis that “John in” is translated to “John em” with the same probability. In another words, the model’s predictions even, without this phrase pair will remain the same.

The example above shows an extreme case, where the event encoded in the phrase pair  $p(\text{John in}, \text{John em})$  is decomposed into independent events, and can be removed without changing the model’s prediction. However, finding and pruning phrase pairs that are independent, based on smaller events is impractical, since most translation events are not strictly independent. However, many phrase pairs can be replaced with derivations using smaller phrases with a small loss in the model’s pre-

<sup>1</sup>For simplicity, we assume at this stage that no reordering model is used

Phrase Pair	Prob	Event
Atomic Phrase Pairs		
in → em	10%	$A_1$
in → na	20%	$A_2$
in → no	20%	$A_3$
in → dentro	5%	$A_4$
in → dentro de	5%	$A_5$
Portugal → Portugal	100%	$B_1$
John → John	100%	$C_1$
Composite Phrase Pairs		
in Portugal → em Portugal	70%	$A_1 \cap B_1$
John in → John em	10%	$C_1 \cap A_1$
John in → John na	20%	$C_1 \cap A_2$
John in → John no	20%	$C_1 \cap A_3$
John in → John dentro	5%	$C_1 \cap A_4$
John in → John dentro de	5%	$C_1 \cap A_5$

Table 1: Phrase Translation Table with associated events

dictions.

Hence, we would like to define a metric for phrase pairs that allows us evaluate how discarding each phrase pair will affect the pruned model’s predictions. By removing phrase pairs that can be derived using smaller phrase pairs with similar probability, it is possible to discard a significant portion of the translation model, while minimizing the impact on the model’s predictions.

### 3.2 Relative Entropy Model for Machine Translation

For each phrase pair  $p_a$ , we define the supporting set  $SP(p_a(s, t)) = S_1, \dots, S_k$ , where each element  $S_i = p_i, \dots, p_j$  is a distinct derivation of  $p_a(s, t)$  that translates  $s$  to  $t$ , with probability  $P(S_i) = P(p_i) \times \dots \times P(p_j)$ . A phrase pair can have multiple elements in its supporting set. For instance, the phrase pair  $p(\text{John in Portugal}, \text{John em Portugal})$ , has 3 elements in the support set:

- $S_1 = \{p(\text{John}, \text{John}), p(\text{in}, \text{em}), p(\text{Portugal}, \text{Portugal})\}$
- $S_2 = \{p(\text{John}, \text{John}), p(\text{in Portugal}, \text{em Portugal})\}$
- $S_3 = \{p(\text{John in}, \text{John em}), p(\text{Portugal}, \text{Portugal})\}$

$S_1$ ,  $S_2$  and  $S_3$  encode 3 different assumptions about the event of translating “John in Portugal” to “John em Portugal”.  $S_1$  assumes that the event is composed by 3 independent events  $A_1$ ,  $B_1$  and  $C_1$ ,  $S_2$  assumes that  $A_1$  and  $B_1$  are dependent, and

groups them into a single composite event  $A_1 \cap B_1$ , which is independent from  $C_1$ , and  $S_3$  groups  $A_1$  and  $C_1$  independently from  $B_1$ . As expected, the event encoded in the phrase pair  $p$  itself is  $A_1 \cap B_1 \cap C_1$ , which assumes that  $A_1$ ,  $B_1$  and  $C_1$  are all dependent. We can see that if any of the events  $S_1$ ,  $S_2$  or  $S_3$  has a “similar probability” as the event coded in the phrase pair, we can remove this phrase pair with a minimal impact in the phrase prediction.

To formalize our notion of “similar probability”, we apply the relative entropy or the Kullback-Leibler divergence, and define the divergence between a pruned translation model  $P_p(s, t)$  and the unpruned model  $P(s, t)$  as:

$$D(P_p||P) = - \sum_{s,t} P(s, t) \log \frac{P_p(t|s)}{P(t|s)} \quad (1)$$

Where  $\frac{P_p(t|s)}{P(t|s)}$ , measures the deviation from the probability emission from the pruned model and the original probability from the unpruned model, for each source-target pair  $s, t$ . This is weighted by the frequency that the pair  $s, t$  is observed, given by  $P(s, t)$ .

Our objective is to minimize  $D(P_p||P)$ , which can be done locally by removing phrase pairs  $p(s, t)$  with the lowest values for  $-P(s, t) \log \frac{P_p(t|s)}{P(t|s)}$ . Ideally, we would want to minimize the relative entropy for all possible source and target sentences, rather than all phrases in our model. However, minimizing such an objective function would be intractable due to reordering, since the probability assigned to a phrase pair in a sentence pair by each model would depend on the positioning of all other phrase pairs used in the sentence. Because of these dependencies, we would not be able to reduce this problem to a local minimization problem. Thus, we assume that all phrase pairs have the same probability regardless of their context in a sentence.

Thus, our pruning algorithm takes a threshold  $\delta$  and prunes all phrase pairs that fail to meet the following criteria:

$$-P(s, t) \log \frac{P_p(t|s)}{P(t|s)} > \delta \quad (2)$$

The main components of this function is the ratio between the emission from the pruned model and

unpruned models given by  $\frac{P_p(t|s)}{P(t|s)}$ , and the weight given to each  $s, t$  pair given by  $P(s, t)$ . In the remainder of this section, we will focus on how to model each of these components in equation 2.

### 3.3 Computing $P(s, t)$

The term  $P(s, t)$  can be seen as a weighting function for each  $s, t$  pair. There is no obvious optimal distribution to model  $P(s, t)$ . In this work, we apply 2 different distributions for  $P(s, t)$ . First, a uniform distribution, where all phrases are weighted equally. Secondly, a multinomial function defined as:

$$P(s, t) = \frac{N(s, t)}{N} \quad (3)$$

where  $N$  is the number of sentence pairs in the parallel data, and  $N(s, t)$  is the number of sentence pairs where  $s$  was observed in the source sentence and  $t$  was observed in the target sentence. Using this distribution, the model is more biased in pruning phrase pairs with  $s, t$  pairs that do not occur frequently.

### 3.4 Computing $\frac{P_p(t|s)}{P(t|s)}$

The computation of  $\frac{P_p(t|s)}{P(t|s)}$  depends on how the decoder adapts when a phrase pair is pruned from the model. In the case of back-off language models, this can be solved by calculating the difference of the logs between the n-gram estimate and the back-off estimate. However, a translation decoder generally functions differently. In our work, we will assume that the decoding will be performed using a Viterbi decoder, such as MOSES (Koehn et al., 2007), where the translation with the highest score is chosen.

In the example above, where  $s$ ="John in Portugal" and  $t$ ="John em Portugal", the decoder would choose the derivation with the highest probability from  $s$  to  $t$ . Using the unpruned model, the possible derivations are either using phrase  $p(s, t)$  or one element of its support set  $S_1, S_2$  or  $S_3$ . On the other hand, on the pruned model where  $p(s, t)$  does not exist, only  $S_1, S_2$  and  $S_3$  can be used. Thus, given a  $s, t$  pair one of three situations may occur. First, if the probability of the phrase pair  $p(s, t)$  is lower than the highest probability element in  $SP(p(s, t))$ , then both the models will choose that element, in which case,  $\frac{P_p(t|s)}{P(t|s)} = 1$ . This can happen, if we define

features that penalize longer phrase pairs, such as lexical weighting, or if we apply smoothing (Foster et al., 2006). Secondly, if the probability of  $p(s, t)$  is equal to the most likely element in  $SP(p(s, t))$ , regardless of whether the unpruned model chooses to use  $p(s, t)$  or that element, the probability emissions of the pruned and unpruned model will be identical. Thus, for this case  $\frac{P_p(t|s)}{P(t|s)} = 1$ . Finally, if the probability of  $p(s, t)$  is higher than other possible derivations, the unpruned model will choose to emit the probability of  $p(s, t)$ , while the pruned model will emit the most likely element in  $SP(p(s, t))$ . Hence, the probability loss between the 2 models, will be the ratio between the probability of  $p(s, t)$  and the probability of the most likely element in  $SP(p(s, t))$ .

From the example above, we can generalize the function for  $\frac{P_p(t|s)}{P(t|s)}$  as:

$$\frac{\prod_{p' \in \text{argmax}(SP(p(s, t)))} P(p')}{P(p(s, t))} \quad (4)$$

Where  $P(p(s, t))$  denotes the probability of  $p(s, t)$  and  $\prod_{p' \in \text{argmax}(SP(p(s, t)))} P(p')$  the most likely sequence of phrasal translations that translates  $s$  to  $t$ , with the probability equal to the product of all phrase translation probabilities in that sequence.

Replacing in equation 2, our final condition that must be satisfied for keeping a phrase pair is:

$$-P(s, t) \log \frac{\prod_{p' \in \text{argmax}(SP(p(s, t)))} P(p')}{P(p(s, t))} > \delta \quad (5)$$

## 4 Application for Phrase-based Machine Translation

We will now show how we apply our entropy pruning model in the state-of-the-art phrase-based translation system MOSES and describe the problems that need to be addressed during the implementation of this model.

### 4.1 Translation Model

The translation model in Moses is composed by a phrase translation model and a phrase reordering model. The first one models, for each phrase pair  $p(s, t)$ , the probability of translating the  $s$  to  $t$  by combining multiple features  $\phi_i$ , weighted by

$w_i^T$ , as  $P_T(p) = \prod_{i=1}^n \phi_i(p)^{w_i^T}$ . The reordering model is similar, but models the local reordering between  $p$ , given the previous and next phrase according to the target side,  $p_P$  and  $p_N$ , or more formally,  $P_R(p|p_P, p_N) = \prod_{i=1}^m \psi_i(p|p_P, p_P)^{w_i^R}$

## 4.2 Building the Support Set

Essentially, implementing our model is equivalent to calculating the components described in equation 5. These are  $P(s, t)$ ,  $P(p(s|t))$  and  $argmax(SP(p(s, t)))$ . Calculating the uniform distribution and multinomial distributions for  $P(s, t)$  is simple, the uniform distribution just assumes the same value for all  $s$  and  $t$ , and the multinomial distribution can be modeled by extracting counts from the parallel corpora.

Calculating  $P(s|t)$  is also trivial, since it only involves calculating  $P_T(p(s, t))$ , which can be done by retrieving the translation features of  $p$  and applying the weights for each feature.

The most challenging task is to calculate  $argmax(SP(p(s, t)))$ , which is similar to the decoding task in machine translation, where we need to find the best translation  $\hat{t}$  for a sentence  $s$ , that is,  $\hat{t} = argmax_t P(s|t)P(t)$ . In practice, we are not searching in the space of possible translations, but in the space of possible derivations, which are sequences of phrase translations  $p_1(s_1, t_1), \dots, p_n(s_n, t_n)$  that can be applied to  $s$  to generate an output  $t$  with the score given by  $P(t) \prod_{i=1}^n P(s_i, t_i)$ .

Our algorithm to determine  $SP(p(s, t))$  can be described as an adaptation to the decoding algorithm in Moses, where we restrict the search space to the subspace  $SP(p(s, t))$ , that is, our search space is only composed by derivations that output  $t$ , without using  $p$  itself. This can be done using the forced decoding algorithm proposed in (Schwartz, 2008). Secondly, the score of a given translation hypothesis does not depend on the language model probability  $P(t)$ , since all derivations in this search space have the same  $t$ , thus we discard this probability from the score function. Finally, rather than using beam search, we exhaustively search all the search space, to reduce the hypothesis of incurring a search error at this stage. This is possible, since phrase pairs are generally smaller than text (less than 8 words), and because we are constraining the search space to  $t$ , which is an order of magnitude smaller than the reg-

ular search space with all possible translations.

## 4.3 Pruning Algorithm

The algorithm to generate a pruned translation model is shown in 1. We iterate over all phrase pairs  $p_1(s_1, t_1), \dots, p_n(s_n, t_n)$ , decode using our forced decoding algorithm from  $s_i$  to  $t_i$ , to obtain the best path  $S$ . If no path is found then it means that the  $p_i$  is atomic. Then, we prune  $p_i$  based on condition 5.

---

### Algorithm 1 Independence Pruning

---

**Require:** pruning threshold  $\delta$ ,  
 unpruned model  $\{p_1(s_1, t_1), \dots, p_n(s_n, t_n)\}$   
**for**  $p_i(s_i, t_i) \in \{p_1(s_1, t_1), \dots, p_n(s_n, t_n)\}$  **do**  
    $S := argmax(SP(p_i)) \setminus p_i$   
    $score := \infty$   
   **if**  $S \neq \{\}$  **then**  
      $score := -P(s, t) \log \frac{\prod_{p'(s', t') \in S} P(s'|t')}{P(s|t)}$   
   **end if**  
   **if**  $score \leq \delta$  **then**  
      $prune(p_i)$   
   **end if**  
**end for**  
**return** pruned model

---

The main bottle neck in this algorithm is finding  $argmax(SP(p_i))$ . While this appears relatively simple and similar to a document decoding task, the size of our task is on a different order of magnitude, since we need to decode every phrase pair in the translation model, which might not be tractable for large models with millions of phrase pairs. We address this problem in section 5.3.

Another problem with this algorithm is that the decision to prune each phrase pair is made assuming that all other phrase pairs will remain in the model. Thus, there is a chance a phrase pair  $p_1$  is pruned because of a derivation using  $p_2$  and  $p_3$  that leads to the same translation. However, if  $p_3$  also happens to be pruned, such a derivation will no longer be possible. One possible solution to address this problem is to perform pruning iteratively, from the smallest phrase pairs (number of words) and increase the size at each iteration. However, we find this undesirable, since the model will be biased into removing smaller phrase pairs, which are generally more useful, since they can be used in multiple derivation to replace larger phrase pairs. In the example above, the model

would eliminate  $p_3$  and keep  $p_1$ , yet the best decision could be to keep  $p_3$  and remove  $p_1$ , if  $p_3$  is also frequently used in derivations of other phrase pairs. Thus, we leave the problem of finding the best set of phrases to prune as future work.

## 5 Experiments

We tested the performance of our system under two different environments. The first is the small scale DIALOG translation task for IWSLT 2010 evaluation (Paul et al., 2010) using a small corpora for the Chinese-English language pair (henceforth referred to as “IWSLT”). The second one is a large scale test using the complete EUROPARL (Koehn, 2005) corpora for the Portuguese-English language pair, which we will denote by “EUROPARL”.

### 5.1 Corpus

The IWSLT model was trained with 30K training sentences. The development corpus and test corpus were taken from the evaluation dataset in IWSLT 2006 (489 tuning and 500 test sentences with 7 references). The EUROPARL model was trained using the EUROPARL corpora with approximately 1.3M sentence pairs, leaving out 1K sentences for tuning and another 1K sentences for tests.

### 5.2 Setup

In the IWSLT experiment, word alignments were generated using an HMM model (Vogel et al., 1996), with symmetric posterior constraints (V. Graça et al., 2010), using the Geppetto toolkit<sup>2</sup>. This setup was used in the official evaluation in (Ling et al., 2010). For the EUROPARL experiment the word alignments were generated using IBM model 4. In both experiments, the translation model was built using the phrase extraction algorithm (Paul et al., 2010), with commonly used features in Moses (Ex: probability, lexical weighting, lexicalized reordering model). The optimization of the translation model weights was done using MERT tuning (Och, 2003) and the results were evaluated using BLEU-4.

### 5.3 Pruning Setup

Our pruning algorithm is applied after the translation model weight optimization with MERT. We gener-

ate multiple translation models by setting different values for  $\delta$ , so that translation models of different sizes are generated at intervals of 5%. We also run the significance pruning (Johnson and Martin, 2007) algorithm in these conditions.

While the IWSLT translation model has only 88,424 phrase pairs, for the EUROPARL experiment, the translation model was composed by 48,762,372 phrase pairs, which had to be decoded. The average time to decode each phrase pair using the full translation model is 4 seconds per sentence, since the table must be read from disk due to its size. This would make translating 48M phrase pairs unfeasible. To address this problem, we divide the phrase pairs in the translation model into blocks of  $K$  phrase pairs, that are processed separately. For each block, we resort to the approach used in MERT tuning, where the model is filtered to only include the phrase pairs that are used for translating tuning sentences. We filter each block with phrase pairs from  $K$  to  $2K$  with the source sentences  $s_K, \dots, s_{2K}$ . Furthermore, since we are force decoding using the target sentences, we also filter the remaining translation models using the target sentences  $t_K, \dots, t_{2K}$ . We used blocks of 10,000 phrase pairs and each filtered table was reduced to less than 1% of the translation table on average, reducing the average decoding time to 0.03 seconds per sentence. Furthermore, each block can be processed in parallel allowing multiple processes to be used for the task, depending on the resources that are available.

### 5.4 Results

Figure 1 shows the BLEU results for different sizes of the translation model for the IWSLT experiment using the uniform and multinomial distributions for  $P(s, t)$ . We observe that there is a range of values from 65% to 95% where we actually observe improvements caused by our pruning algorithm, with the peak at 85% for the uniform distribution, where we improve from 15.68 to 15.82 (0.9% improvement). Between 26% and 65%, the BLEU score is lower than the baseline at 100%, with the minimum at 26% with 15.54, where only atomic phrase pairs remain and both the multinomial and uniform distribution have the same performance, obviously. This is a considerable reduction in phrase table size by sacrificing 0.14 BLEU points. Regarding the com-

<sup>2</sup><http://code.google.com/p/geppetto/>

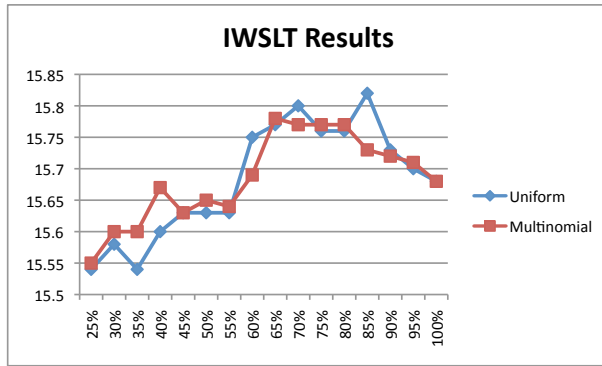


Figure 1: Results for the IWSLT experiment. The x-axis shows the percentage of the phrase table used. The BLEU scores are shown in the y-axis. Two distributions for  $P(s, t)$  were tested Uniform and Multinomial.

comparison between the uniform and multinomial distribution, we can see that both distributions yield similar results, specially when a low number of phrase pairs is pruned. In theory, the multinomial distribution should yield better results, since the pruning model will prefer to prune phrase pairs that are more likely to be observed. However, longer phrase pairs, which tend compete with other long phrase pairs on which get pruned first. These phrase pairs generally occur only once or twice, so the multinomial model will act similarly to the uniform model regarding longer phrase pairs. On the other hand, as the model size reduces, we can see that using multinomial distribution seems to start to improve over the uniform distribution.

The comparison between our pruning model and pruning based on significance is shown in table 2. These models are hard to compare, since not all phrase table sizes can be obtained using both metrics. For instance, the significance metric can either keep or remove all phrase pairs that only appear once, leaving a large gap of phrase table sizes that cannot be attained. In the EUROPARL experiment the sizes of the table suddenly drops from 60% to 8%. The same happens with our metric that cannot distinguish atomic phrase pairs. In the EUROPARL experiment, we cannot generate phrase tables with sizes smaller than 15%. Thus, we only show results at points where both algorithms can produce a phrase table.

Significant improvements are observed in the

Table size	Significance Pruning	Entropy (u) Pruning	Entropy (m) Pruning
<b>IWSLT</b>			
57K (65%)	14.82	15.77	<b>15.78</b>
71K (80%)	15.14	15.76	<b>15.77</b>
80K (90%)	15.31	<b>15.73</b>	15.72
88K (100%)	15.68	15.68	15.68
<b>EUROPARL</b>			
29M (60%)	28.64	28.82	<b>28.91</b>
34M (70%)	28.84	28.94	<b>28.99</b>
39M (80%)	28.86	<b>28.99</b>	<b>28.99</b>
44M (90%)	28.91	29.00	<b>29.02</b>
49M (100%)	29.18	29.18	29.18

Table 2: Comparison between Significance Pruning (Significance Pruning) and Entropy-based pruning using the uniform (Entropy (u) Pruning) and multinomial distributions (Entropy (m) Pruning).

IWSLT experiment, where significance pruning does not perform as well. On the other hand, on the EUROPARL experiment, our model only achieves slightly higher results. We believe that this is related by the fact the EUROPARL corpora is generated from automatically aligning documents, which means that there are misaligned sentence pairs. Thus, many spurious phrase pairs are extracted. Significance pruning performs well under these conditions, since the measure is designed for this purpose. In our metric, we do not have any means for detecting spurious phrase pairs, in fact, spurious phrase pairs are probably kept in the phrase table, since each distinct spurious phrase pair is only extracted once, and thus, they have very few derivations in its support set. This suggests, that the significance score can be integrated in our model to improve our model, which we leave as future work.

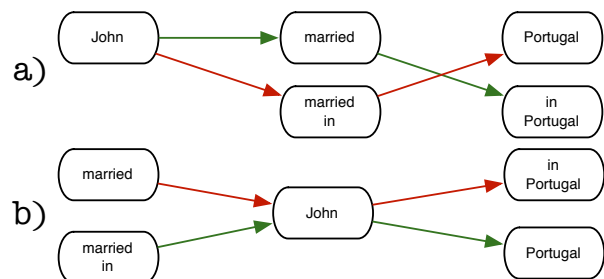


Figure 2: Translation order in for different reordering starting from left to right.

We believe that in language pairs such as Chinese-



English with large distance reorderings between phrases are more prone to search errors and benefit more from our pruning algorithm. To illustrate this, let us consider the source sentence “John married in Portugal”, and translating either using the blocks “John”, “married” and “in Portugal” or the blocks “John”, “married in”, “Portugal”, the first hypothesis would be much more viable, since the word “Portugal” is more relevant as the context for the word “in”. Thus, the key choice for the decoder is to decide whether to translate using “married” with or without “in”, and it is only able to predict that it is better to translate “married” by itself until it finds that “in” is better translated with “Portugal”. Thus, a search error occurs if the hypothesis where “married” is translated by itself is removed. In figure 2, we can see the order that blocks are considered for different reorderings, starting from left to right. In a), we illustrate the case for a monotonous translation. We observe that the correct decision between translating “married in” or just “married” is found immediately, since the blocks “Portugal” and “in Portugal” are considered right afterwards. In this case, it is unlikely that the hypothesis using “married” is removed. However, if we consider that due to reordering, “John” is translated after “married” and before “Portugal”, which is shown in b). Then, the correct decision can only be found after considering “John”. In this case, “John” does not have many translations, so the likelihood of eliminating the correct hypothesis. However, if there were many translations for John, it is highly likely that the correct partial hypothesis is eliminated. Furthermore, the more words exist between “married” and “Portugal”, the more likely will the correct hypothesis not exist when we reach “Portugal”. By pruning the hypothesis “married in” a priori, we contribute in preventing such search errors.

We observe that some categories of phrase pairs that are systematically pruned, but these cannot be generalized in rules, since there are many exceptions. The most obvious type of phrase pairs are phrases with punctuations, such as “谢谢.” to “thanks .” and “. 谢谢” to “thanks .”, since “.” is translated independently from most contextual words. However, this rule should not be generalized, since in some cases “.” is a relevant contextual marker. For instance, the word “please” is translated

to “请” in the sentence ‘open the door, please.’ and translated to “使高兴” in “please my advisors”. Another example are sequences of numbers, which are generally translated literally. For instance, “八(8) 三(3) 八(8)” is translated to “eight three eight” (Ex: “room eight three eight”). Thus, phrase pairs for number sequences can be removed, since those numbers can be translated one by one. However, for sequences such as “一(1) 八(8)”, we need a phrase pair to represent this specifically. This is because “一(1)” can be translated to “one”, but also to “a”, “an”, “single”. Other exceptions include “一(1) 一(1)”, which tends to be translated as “eleven”, and which tends to be translated to “o”, rather than “zero” in sequences (“room eleven o five”).

## 6 Conclusions

We present a pruning algorithm for Machine Translation based on relative entropy, where we assess whether the translation event encoded in a phrase pair can be decomposed into combinations of events encoded in other phrase pairs. We show that such phrase pairs can be removed from the translation model with little negative impact or even a positive one in the overall translation quality. Tests show that our method yields comparable or better results with state of the art pruning algorithms.

As future work, we would like to combine our approach with significance pruning, since both approaches are orthogonal and address different issues. We also plan to improve the pruning step of our algorithm to find the optimal set of phrase pairs to prune given the pruning threshold.

The code used in this work will be made available.

## 7 Acknowledgements

This work was partially supported by FCT (INESC-ID multiannual funding) through the PIDDAC Program funds, and also through projects CMU-PT/HuMach/0039/2008 and CMU-PT/0005/2007. The PhD thesis of Wang Ling is supported by FCT grant SFRH/BD/51157/2010. The authors also wish to thank the anonymous reviewers for many helpful comments.

## References

- Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.*, 19:263–311, June.
- George Foster, Roland Kuhn, and Howard Johnson. 2006. Phrasetable smoothing for statistical machine translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP '06*, pages 53–61, Stroudsburg, PA, USA. Association for Computational Linguistics.
- J Howard Johnson and Joel Martin. 2007. Improving translation quality by discarding most of the phrasetable. In *In Proceedings of EMNLP-CoNLL'07*, pages 967–975.
- Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03*, pages 48–54, Morristown, NJ, USA. Association for Computational Linguistics.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-burch, Richard Zens, Rwth Aachen, Alexandra Constantin, Marcello Federico, Nicola Bertoldi, Chris Dyer, Brooke Cowan, Wade Shen, Christine Moran, and Ondrej Bojar. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.
- Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.
- Wang Ling, Tiago Luís, João Graça, Luísa Coheur, and Isabel Trancoso. 2010. Towards a general and extensible phrase-extraction algorithm. In *IWSLT '10: International Workshop on Spoken Language Translation*, pages 313–320, Paris, France.
- Stephen Vogal Matthias Eck and Alex Waibel. 2007. Estimating phrase pair relevance for translation model pruning. *MTSummit XI*.
- Robert C. Moore and Chris Quirk. 2009. Less is more: significance-based n-gram selection for smaller, better language models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2, EMNLP '09*, pages 746–755, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, ACL '03*, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Michael Paul, Marcello Federico, and Sebastian Stüker. 2010. Overview of the iwslt 2010 evaluation campaign. In *IWSLT '10: International Workshop on Spoken Language Translation*, pages 3–27.
- Lane Schwartz. 2008. Multi-source translation methods. In *Proceedings of AMTA*, pages 279–288.
- Kristie Seymore and Ronald Rosenfeld. 1996. Scalable backoff language models. In *In Proceedings of ICSLP*, pages 232–235.
- Andreas Stolcke. 1998. Entropy-based pruning of backoff language models. In *In Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pages 270–274.
- Nadi Tomeh, Nicola Cancedda, and Marc Dymetman. 2009. Complexity-based phrase-table filtering for statistical machine translation. *MTSummit XII*, Aug.
- João V. Graça, Kuzman Ganchev, and Ben Taskar. 2010. Learning Tractable Word Alignment Models with Complex Constraints. *Comput. Linguist.*, 36:481–504.
- S. Vogel, H. Ney, and C. Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics.