Model With Minimal Translation Units, But Decode With Phrases

Nadir Durrani*
University of Edinburgh
dnadir@inf.ed.ac.uk

Alexander Fraser **Helmut Schmid**
University of Stuttgart
fraser,schmid@ims.uni-stuttgart.de

Abstract

N-gram-based models co-exist with their phrase-based counterparts as an alternative SMT framework. Both techniques have pros and cons. While the N-gram-based framework provides a better model that captures both source and target contexts and avoids spurious phrasal segmentation, the ability to memorize and produce larger translation units gives an edge to the phrase-based systems during decoding, in terms of better search performance and superior selection of translation units. In this paper we combine N-gram-based modeling with phrase-based decoding, and obtain the benefits of both approaches. Our experiments show that using this combination not only improves the search accuracy of the N-gram model but that it also improves the BLEU scores. Our system outperforms state-of-the-art phrase-based systems (Moses and Phrasal) and N-gram-based systems by a significant margin on German, French and Spanish to English translation tasks.

1 Introduction

Statistical Machine Translation advanced from word-based models (Brown et al., 1993) towards more sophisticated models that take contextual information into account. Phrase-based (Och and Ney, 2004; Koehn et al., 2003) and N-gram-based (Mariño et al., 2006) models are two instances of such frameworks. While the two models have some common properties, they are substantially different.

Phrase-based systems employ a simple and effective machinery by learning larger chunks of translation called phrases[1]. Memorizing larger units enables the phrase-based model to learn local dependencies such as short reorderings, idioms, insertions and deletions, etc. The model however, has the following drawbacks: i) it makes independence assumptions over phrases ignoring the contextual information outside of phrases ii) it has issues handling long-distance reordering iii) it has the spurious phrasal segmentation problem which allows multiple derivations of a bilingual sentence pair having different model scores for each segmentation.

Modeling with minimal translation units helps address some of these issues. The N-gram-based SMT framework is based on tuples. Tuples are minimal translation units composed of source and target cepts[2]. N-gram-based models are Markov models over sequences of tuples (Mariño et al., 2006; Crego and Mariño, 2006) or operations encapsulating tuples (Durrani et al., 2011). This mechanism has several useful properties. Firstly, no phrasal independence assumption is made. The model has access to both source and target context outside of phrases. Secondly the model learns a unique derivation of a bilingual sentence given its alignment, thus avoiding the spurious segmentation problem.

Using minimal translation units, however, results in a higher number of search errors because of i)

*Much of the work presented here was carried out while the first author was at the University of Stuttgart.

[1] A phrase-pair in PBSMT is a sequence of source and target words that is translation of each other, and is not necessarily a linguistic constituent. Phrases are built by combining minimal translation units and ordering information.

[2] A cept is a group of words in one language that is translated as a minimal unit in one specific context (Brown et al., 1993).

Proceedings of NAACL-HLT 2013, pages 1–11,
Atlanta, Georgia, 9–14 June 2013.

poor translation selection, ii) inaccurate future-cost estimates and iii) incorrect early pruning of hypotheses that would produce better model scores if allowed to continue. In order to deal with these problems, search is carried out only on a graph of pre-calculated orderings, and ad-hoc reordering limits are imposed to constrain the search space (Crego et al., 2005; Crego and Mariño, 2006), or a higher beam size is used in decoding (Durrani et al., 2011). The ability to memorize and produce larger translation chunks during decoding, on the other hand, gives a distinct advantage to the phrase-based system during search. Phrase-based systems i) have access to uncommon translations, ii) do not require higher beam sizes, iii) have more accurate future-cost estimates because of the availability of phrase-internal language model context before search is started. To illustrate this consider the German-English phrase-pair "schoß ein Tor – scored a goal", composed from the tuples (cept-pairs) "schoß – scored", "ein – a" and "Tor – goal". It is likely that the N-gram system does not have the tuple "schoß – scored" in its n-best translation options because "scored" is an uncommon translation for "schoß" outside the sports domain. Even if "schoß – scored" is hypothesized, it will be ranked quite low in the stack until "ein" and "Tor" are generated in the next steps. A higher beam is required to prevent it from getting pruned. Phrase-based systems, on the other hand, are likely to have access to the phrasal unit "schoß ein Tor – scored a goal" and can generate it in a single step. Moreover, a more accurate future-cost estimate can be computed because of the available context internal to the phrase.

In this work, we extend the N-gram model, based on operation sequences (Durrani et al., 2011), to use phrases during decoding. The main idea is to study whether a combination of modeling with minimal translation units and using phrasal information during decoding helps to solve the above-mentioned problems.

The remainder of this paper is organized as follows. The next two sections review phrase-based and N-gram-based SMT. Section 2 provides a comparison of phrase-based and N-gram-based SMT. Section 3 summarizes the operation sequence model (OSM), the main baseline for this work. Section 4 analyzes the search problem when decoding with

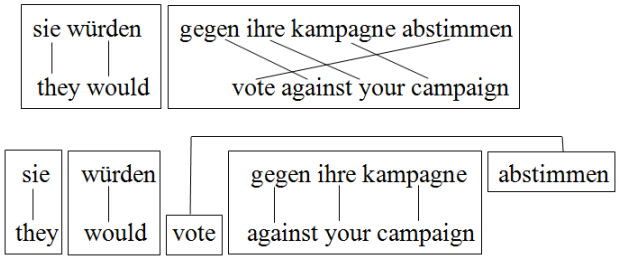


Figure 1: Different Segmentations of a Bilingual Sentence Pair

minimal units. Section 5 discusses how information available in phrases can be used to improve search performance. Section 6 presents the results of this work. We conducted experiments on the German-to-English and French-to-English translation tasks and found that using phrases in decoding improves both search accuracy and BLEU scores. Finally we compare our system with two state-of-the-art phrase-based systems (Moses and Phrasal) and two state-of-the-art N-gram-based systems (Ncode and OSM) on standard translation tasks.

2 Previous Work

Phrase-based and N-gram-based SMT are alternative frameworks for string-to-string translation. Phrase-based SMT segments a bilingual sentence pair into phrases that are continuous sequences of words (Och and Ney, 2004; Koehn et al., 2003) or discontinuous sequences of words (Galley and Manning, 2010). These phrases are then reordered through a lexicalized reordering model that takes into account the orientation of a phrase with respect to its previous phrase (Tillmann and Zhang, 2005) or block of phrases (Galley and Manning, 2008).

There are several drawbacks of the phrase-based model. Firstly it makes an independence assumption over phrases, according to which phrases are translated independently of each other, thus ignoring the contextual information outside of the phrasal boundary. This problem is corrected by the monolingual language model that takes context into account. But often the language model cannot compensate for the dispreference of the translation model for non-local dependencies. The second problem is that the model is unaware of the actual phrasal segmentation of a sentence during training. It therefore learns all possible ways of segmenting a bilingual sentence. Different segmentations of a bilingual sentence re-

sult in different probability scores for the translation and reordering models, causing spurious ambiguity in the model. See Figure 1. In the first segmentation, the model learns the lexical and reordering probabilities of the phrases "sie würden – they would" and "gegen ihre kampagne abstimmen – vote against your campaign". In the second segmentation, the model learns the lexical and reordering probabilities of the phrases "sie – they" "würden – would", "abstimmen – vote", "gegen ihre kampagne – against your campaign". Both segmentations result in different translation and reordering scores. This kind of ambiguity in the model subsequently results in the presence of many different equivalent segmentations in the search space. Also note that the two segmentations contain different information. From the first segmentation the model learns the dependency between the verb "abstimmen – vote" and the phrase "gegen ihre kampagne – against your campaign". The second segmentation allows the model to capture the reordering of the complex verb predicate "würden – would" and "abstimmen – vote" by learning that the verb "abstimmen – vote" is discontinuous with respect to the auxiliary. This information cannot be captured in the first segmentation because of the phrasal independence assumption and stiff phrasal boundaries. The model loses one of the dependencies depending upon which segmentation it chooses during decoding.

N-gram-based SMT is an instance of a joint model that generates source and target strings together in bilingual translation units called tuples. Tuples are essentially phrases but they are atomic units that cannot be decomposed any further. This condition of atomicity results in a unique segmentation of the bilingual sentence pair given its alignments. The model does not make any phrasal independence assumption and generates a tuple by looking at a context of $n-1$ previous tuples (or operations). This allows the N-gram model to model all the dependencies through a single derivation.

The main drawback of N-gram-based SMT is its poor search mechanism which is inherent from using minimal translation units during search. Decoding with tuples has problems with a high number of search errors caused by lower translation coverage, inaccurate future-cost estimation and pruning of correct hypotheses (see Section 4.2 for details). Crego and Mariño (2006) proposed a way to couple reordering and search through POS-based rewrite rules. These rules are learned during training when units with crossing alignments are unfolded through source linearization to form minimal tuples. For example, in Figure 1, the N-gram-based MT will linearize the word sequence "gegen ihre kampagne abstimmen" to "abstimmen gegen ihre kampagne", so that it is in the same order as the English words. It also learns a POS-rule "IN PRP NN VB $\rightarrow$ VB IN PRP NN". The POS-based rewrite rules serve to precompute the orderings that are hypothesized during decoding. Coupling reordering and search allows the N-gram model to arrange hypotheses in 2^m stacks (for an m word source sentence), each containing hypotheses that cover exactly the same foreign words. This removes the need for future-cost estimation[3]. Secondly, memorizing POS-based rules enables phrase-based like reordering, however without lexical selection. There are three drawbacks of this approach. Firstly, lexical generation and reordering are decoupled. Search is only performed on a small number of reorderings, pre-calculated using the source side and completely ignoring the target-side. And lastly, the POS-based rules face data sparsity problems especially in the case of long distance reorderings.

Durrani et al. (2011) recently addressed these problems by proposing an operation sequence N-gram model which strongly couples translation and reordering, hypothesizes all possible reorderings and does not require POS-based rules. Representing bilingual sentences as a sequence of operations enables them to memorize phrases and lexical reordering triggers like PBSMT. However, using minimal units during decoding and searching over all possible reorderings means that hypotheses can no longer be arranged in 2^m stacks. The problem of inaccurate future-cost estimates resurfaces resulting in more search errors. A higher beam size of 500 is therefore used to produce translation units in comparison to phrase-based systems. This, however, still does not eliminate all search errors. This paper shows that using phrases instead of cepts in de-

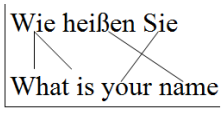
[3]Using m stacks with future-cost estimation is a more efficient solution but is not used "due to the complexity of accurately computing these estimations in the N-gram architecture" (Crego et al., 2011).

coding improves the search accuracy and translation quality. It also shows that using some phrasal information in cept-based decoding captures some of these improvements.

3 Operation Sequence Model

The N-gram model with integrated reordering models a *sequence of operations* obtained through the transformation of a bilingual sentence pair. An operation can either be to i) generate a sequence of source and target words, ii) to insert a gap as a placeholder for skipped words, iii) or to jump forward and backward in a sentence to translate words discontinuously. The translate operation Generate(X,Y) encapsulates the translation tuple (X,Y). It generates source and target translations simultaneously[4]. This is similar to N-gram-based SMT except that the tuples in the N-gram-based model are generated monotonically, whereas in this case lexical generation and reordering information is strongly coupled in an operation sequence.

Consider the phrase pair:



The model memorizes it through the sequence:

Generate(Wie, What is) → Gap → Generate (Sie, your) → Jump Back (1) → Generate (heissen, name)

Let $O = o_1, \ldots, o_{j-1}$ be a sequence of operations as hypothesized by the translator to generate the bilingual sentence pair $\langle F, E \rangle$ with an alignment function A. The translation model is defined as:

$$p(F, E, A) = p(o_1^J) = \prod_{j=1}^{J} p(o_j | o_{j-n+1}, ..., o_{j-1})$$

where n indicates the amount of context used. The translation model is implemented as an N-gram model of operations using SRILM-Toolkit (Stolcke, 2002) with Kneser-Ney smoothing. A 9-gram model is used. Several count-based features such as gap and open gap penalties and distance-based features such as gap-width and reordering distance are added to the model, along with the lexical weighting and length penalty features in a standard log-linear framework (Durrani et al., 2011).

[4]The generation is carried out in the order of the target language E.

4 Search

4.1 Overview of Decoding Framework

The decoding framework used in the operation sequence model is based on Pharaoh (Koehn, 2004a). The decoder uses beam search to build up the translation from left to right. The hypotheses are arranged in m stacks such that stack i maintains hypotheses that have already translated i many foreign words. The ultimate goal is to find the best scoring hypothesis, that has translated all the words in the foreign sentence. The overall process can be roughly divided into the following steps: i) extraction of translation units ii) future-cost estimation, iii) hypothesis extension iv) recombination and pruning.

During the hypothesis extension each extracted phrase is translated into a sequence of operations. The reordering operations (gaps and jumps) are generated by looking at the position of the translator, the last foreign word generated etc. (Refer to Algorithm 1 in Durrani et al. (2011)). The probability of an operation depends on the $n-1$ previous operations. The model backs-off to the smaller n-grams of operations if the full history is unknown. We use Kneser-Ney smoothing to handle back-off[5].

4.2 Drawbacks of Cept-based Decoding

One of the main drawbacks of the operation sequence model is that it has a more difficult search problem than the phrase-based model. The operation model, although based on minimal translation units, can learn larger translation chunks by memorizing a sequence of operations. However, using cepts during decoding has the following drawbacks: i) the cept-based decoder does not have access to all the translation units that a phrase-based decoder uses as part of a larger phrase. ii) it requires a higher beam size to prevent early pruning of better hypotheses that lead toward higher model scores when allowed to continue and iii) it uses worse future-cost estimates than the phrase-based decoder.

Recall the example from the last section. For the cept-based decoder to generate the same phrasal translation, it requires three separate tuple translations "Wie – what is", "Sie – your" and "heißen – name". Here we are faced with three challenges.

[5]We also tried Witten-Bell and Good Turing methods of discounting and found Kneser-Ney smoothing to produce the best results.

Translation Coverage: The first problem is that the N-gram model does not have the same coverage of translation options. The English cepts "what is", "your" and "name" are not good candidate translations for the German cepts "Wie", "Sie" and "heißen", respectively. When extracting tuple translations for these cepts from the Europarl data for our system, the tuple "Wie – what is" is ranked 124^{th}, "heißen – name" is ranked 56^{th}, and "Sie – your" is ranked 9^{th} in the list of n-best translation candidates. Typically only the 20 best translation options are used, to reduce the decoding time, and such phrasal units with less frequent cept translations are never hypothesized in the N-gram-based systems. The phrase-based system on the other hand can extract the phrase "Wie heißen Sie – what is your name" even if it is observed only once during training. A similar problem is also reported in Costa-jussà et al. (2007). When trying to reproduce the sentences in the n-best translation output of the phrase-based system, the N-gram-based system was only able to produce 37.5% of the sentences in the Spanish-to-English and 37.2% in the English-to-Spanish translation tasks. In comparison the phrase-based system was able to reproduce 57.5% and 48.6% of the sentences in the n-best translation output of the Spanish-to-English and English-to-Spanish N-gram-based systems.

Larger Beam Size: A related problem is that a higher beam size is required in cept-based decoding to prevent uncommon translations from getting pruned. The phrase-based system can generate the phrase-pair "Wie heißen Sie – what is your name" in a single step placing it directly into the stack three words to the right. The cept-based decoder generates this phrase in three stacks with the tuple translations "Wie – What is", "Sie – your" and "heißen – name". A very large stack size is required during decoding to prevent the pruning of "Wie – What is" which is ranked quite low in the stack until the tuple "Sie – your" is hypothesized in the next stack. Costa-jussà et al. (2007) reports a significant drop in the performance of N-gram-based SMT when a beam size of 10 is used instead of 50 in their experiments. For the (cept-based) operation sequence model, Durrani et al. (2011) required a stack size of 500. In comparison, the translation quality achieved by phrase-based SMT remains the same when varying the beam size between 5 and 50.

Future-Cost Estimation: A third problem is caused by inaccurate future-cost estimation. Using phrases helps phrase-based SMT to better estimate the future language model cost because of the larger context available, and allows the decoder to capture local (phrase-internal) reorderings in the future cost. In comparison the future cost for tuples is mostly unigram probabilities. The future-cost estimate for the phrase pair "Wie heißen Sie – What is your name" is estimated by calculating the cost of each feature. The language model cost, for example, is estimated in the phrase-based system as follows:

$$\begin{aligned} p_{lm} = {} & p(\text{What}) \times p(\text{is}|\text{What}) \times p(\text{your}|\text{What is}) \\ & \times p(\text{name}|\text{What is your}) \end{aligned}$$

The cost of the direct phrase translation probability, one of the features used in the phrase translation model, is estimated as:

$$p_{tm} = p(\text{What is your name}|\text{Wie heißen Sie})$$

Phrase-based SMT is aware during the preprocessing step that the words "Wie heißen Sie" may be translated as a phrase. This is helpful for estimating a more accurate future cost because the phrase-internal context is already available. The same is not true for the operation sequence model, to which only minimal units are available. The operation model does not have the information that "Wie heißen Sie" may be translated as a phrase during decoding. The future-cost estimate available to the operation model for the span covering "Wie heißen Sie" will have unigram probabilities for both the translation and language model:

$$p_{lm} = p(\text{What}) \times p(\text{is}|\text{What}) \times p(\text{your}) \times p(\text{name})$$

$$\begin{aligned} p_{tm} = {} & p(\text{Generate(Wie, What is)}) \times p(\text{Generate} \\ & (\text{heißen,name})) \times p(\text{Generate(Sie, your)}) \end{aligned}$$

Thus the future-cost estimate in the operation model is much worse than that of the phrase-based model. The poor future-cost estimation leads to search errors, causing a drop in the translation quality. A more accurate future-cost estimate for the translation model cost would be:

$$p_{tm} = \text{p(Generate(Wie,What is))} \times \text{p(Insert Gap|C)} \times \text{p(Generate(Sie,your)|C)} \times \text{p(Jump Back(1)|C)} \text{ p(Generate(heißen,name)|C)}$$

where C is the context, i.e., the $n-1$ previously generated operations. The future-cost estimates computed in this manner are much more accurate because they not only consider context, but also take the reordering operations into account.

5 N-gram Model with Phrase-based Decoding

In the last section we discussed the disadvantages of using cepts during search in a left-to-right decoding framework. We now define a method to empirically study the mentioned drawbacks and whether using information available in phrase-pairs during decoding can help improve search accuracy and translation quality.

5.1 Training

We extended the training steps in Durrani et al. (2011) to extract a phrase lexicon from the parallel data. We extract all phrase pairs of length 6 and below, that are consistent (Och et al., 1999) with the word alignments. Only continuous phrases as used in a traditional phrase-based system are extracted thus allowing only inside-out (Wu, 1997) type of alignments. The future cost of each feature component used in the log-linear model is calculated. The operation sequence required to hypothesize each phrase is generated and its future cost is calculated. The future costs of other features such as language models, lexicalized probability features, etc. are also estimated. The estimates of the count-based reordering penalties (gap penalty and open gap penalty) and the distance-based features (gap-width and reordering distance) could not be estimated previously with cepts but are available when using phrases.

5.2 Decoding

We extended the decoder developed by Durrani et al. (2011) and tried three ideas. In our primary experiments we enabled the decoder to use phrases instead of cepts. This allows the decoder to i) use phrase-internal context when computing the future-cost estimates, ii) hypothesize translation options not available to the cept-based decoder iii) cover multiple source words in a single step subsequently improving translation coverage and search. Note that using phrases instead of cepts during decoding, does not reintroduce the spurious phrasal segmentation problem as is present in the phrase-based system, because the model is built on minimal units which avoids segmentation ambiguity. Different compositions of the same phrasal unit lead to exactly the same model score. We therefore do not create any alternative compositions of the same phrasal unit during decoding. This option is not available in phrase-based decoding, because an alternative composition may lead towards a better model score.

In our secondary set of experiments, we used cept-based decoding but modified the decoder to use information available from the phrases extracted for the test sentences. Firstly, we used future-cost estimates from the extracted phrases (see system cept.500.fc in Table1). This however, leads to inconsistency in the cases where the future cost is estimated from some phrasal unit that cannot be generated through the available cept translations. For example, say the best cost to cover the sequence "Wie heißen Sie" is given by the phrase "What is your name". The 20-best translation options in cept-based system, however, do not have tuples "Wie – What" and "heißen – name". To remove this discrepancy, we add all such tuples that are used in the extracted phrases, to the list of extracted cepts (system cept.500.fc.t). We also studied how much gain we obtain by only adding tuples from phrases and using cept-based future-cost estimates (system cept.500.t).

5.3 Evaluation Method

To evaluate our modifications we apply a simple strategy. We hold the model constant and change the search to use the baseline decoder, which uses minimal translation units, or the modified decoders that use phrasal information during decoding. The model parameters are optimized by running MERT (minimum error rate training) for the baseline decoder on the dev set. After we have the optimized weights, we run the baseline decoder and our modifications on the test. Note that because all the decoding runs use the same feature vector, the model

stays constant, only search changes. This allows us to compare different decoding runs, obtained using the same parameters, but different search strategies, in terms of model scores. We compute a search accuracy and translation quality for each run.

Search accuracy is computed by comparing translation hypotheses from the different decoding runs. We form a collection of the best scoring hypotheses by traversing through all the runs and selecting the sentences with highest model score. For each input sentence we select a single best scoring hypothesis. The best scoring hypothesis can be contributed from several runs. In this case all these runs will be given a credit for that particular sentence when computing the search accuracy. The search accuracy of a decoding run is defined as the percentage of hypotheses that were contributed from this run, when forming a list of best scoring hypotheses. For example, for a test set of 1000 sentences, the accuracy of a decoding run would be 30% if it was able to produce the best scoring hypothesis for 300 sentences in the test set. Translation quality is measured through BLEU (Papineni et al., 2002).

6 Experimental Setup

We initially experimented with two language pairs: German-to-English (G-E) and French-to-English (F-E). We trained our system and the baseline systems on most of the data[6] made available for the translation task of the *Fourth Workshop on Statistical Machine Translation*.[7] We used 1M bilingual sentences, for the estimation of the translation model and 2M sentences from the monolingual corpus (news commentary) which also contains the English part of the bilingual corpus. Word alignments are obtained by running GIZA++ (Och and Ney, 2003) with the grow-diag-final-and (Koehn et al., 2005) symmetrization heuristic. We follow the training steps described in Durrani et al. (2011), consisting of i) post-processing the alignments to remove discontinuous and unaligned target cepts, ii) conversion of bilingual alignments into operation sequences, iii) estimation of the n-gram language models.

[6]We did not use all the available data due to scalability issues. The scores reported are therefore well below those obtained by the systems submitted to the WMT evaluation series.

[7]http://www.statmt.org/wmt09/translation-task.html

6.1 Search Accuracy Results

We divided our evaluation into two halves. In the first half we carried out experiments to measure search accuracy and translation quality of our decoders against the baseline cept-based OSM (cept.500) that uses minimal translation units with a stack size of 500. We used the version of the cept-based OSM system that does not allow discontinuous[8] source cepts. To increase the speed of the system we used a hard reordering limit of 15[9], in the baseline decoder and our modifications, disallowing jumps that are beyond 15 words from the first open gap. For each extracted cept or phrase 10-best translation options are extracted.

Using phrases in search reduces the decoding speed. In order to make a fair comparison, both the phrase-based and the baseline cept-based decoders should be allowed to run for the same amount of time. We therefore reduced the stack size in the phrase-based decoder so that it runs in the same amount of time as the cept-based decoder. We found that using a stack size of 200[10] for the phrase-based decoder was comparable in speed to using a stack-size of 500 in the cept-based decoding.

We first tuned the baseline on dev[11] to obtain an optimized weight vector. We then ran the baseline and our decoders as discussed in Section 5.2 on the dev-test. Then we repeated this experiment by tuning the weights with our phrase-based decoder (using a stack size of 100) and ran all the decoders again using the new weights.

Table 1 shows the average search accuracies and BLEU scores of the two experiments. Using phrases during decoding in the G-E experiments resulted in a statistically significant[12] 0.69 BLEU points gain comparing our best system phrase.200 with the baseline system cept.500. We mark a result as sig-

[8]Discontinuous source-side units did not lead to any improvements in (Durrani et al., 2011) and increased the decoding times by multiple folds. We also found these to be less useful.

[9]Imposing a hard reordering limit significantly reduced the decoding time and also slightly increased the BLEU scores.

[10]Higher stack sizes leads to improvement in model scores for both German-English and French-English and slight improvement of BLEU in the case of the former.

[11]We used news-dev2009a as dev and news-dev2009b as dev-test and tuned the weights with Z-MERT (Zaidan, 2009).

[12]We use bootstrap resampling (Koehn, 2004b) to test our results against the baseline result.

System	German		French	
	Acc.	BLEU	Acc.	BLEU
Baseline System cept.stack-size				
cept.50	25.95%	19.50	42.10%	21.44
cept.100	30.04%	19.79	47.32%	21.70
cept.200	35.17%	19.98	51.47%	21.82
cept.500	41.56%	20.14	54.93%	21.87
Our Cept-based Decoders				
cept.500.fc	48.44%	20.52*	54.73%	21.86
cept.500.t	52.24%	20.34	67.95%	22.00
cept.500.fc.t	61.81%	20.53*	67.76%	21.96
Our Phrase-based Decoders				
phrase.50	58.88%	20.58*	80.83%	22.04
phrase.100	69.85%	20.73*	88.34%	22.13
phrase.200	79.71%	20.83*	92.93%	22.17*

Table 1: Search Accuracies (Acc.) and BLEU scores of the Baseline and Our Decoders with different Stack Sizes (fc = Future Cost Estimated from Phrases, t = Cept Translation Options enriched from Phrases)

nificant if the improvement shown by our decoder over the baseline decoder (cept.500) is significant at the $p \leq 0.05$ level, in both the runs. All the outputs that show statistically significant improvements over the baseline decoder (cept.500) in Table 1 are marked with an asterisk.

The search accuracy of our best system (phrase.200), in G-E experiments is roughly 80%, which means that 80% of the times the phrase-based decoder (using stack size 200) was able to produce the same model score or a better model score than the cept-based decoders (using a stack size of 500). Our F-E experiments also showed improvements in BLEU and model scores. The search accuracy of our best system phrase.200 is roughly 93% as compared with 55% in the baseline decoder (cept.500) giving a BLEU point gain of +0.30 over the baseline.

Our modifications to the cept-based decoder also showed improvements. We found that extending the cept translation table (cept.500.t) using phrases helps both in G-E and F-E experiments by extending the list of n-best translation options by 18% and 18.30% respectively. Using future costs estimated from phrases (cept.500.fc) improved both search accuracy and BLEU scores in G-E experiments, but does not lead to any improvements in the F-E experiments, as both BLEU and model scores drop slightly. We looked at a few examples where the model score dropped and found that in these cases, the best scoring hypotheses are ranked very low earlier in the decoding and make their way to the top gradually in subsequent steps. A slight difference in the future-cost estimate prunes these hypotheses in one or the other decoder. We found future cost to be more critical in G-E than F-E experiments. This can be explained by the fact that more reordering is required in G-E and it is necessary to account for the reordering operations and jump-based features (gap-based penalties, reordering distance and gap-width) in the future-cost estimation. F-E on the other hand is largely monotonic except for a few short distance reorderings such as flipping noun and adjective.

6.2 Comparison with other Baseline Systems

In the second half of our evaluation we compared our best system phrase.200 with the baseline system cept.500, and other state-of-the-art phrase-based and N-gram-based systems on German-to-English, French-to-English, and Spanish-to-English tasks[13]. We used the official evaluation data (news-test sets) from the Statistical Machine Translation Workshops 2009-2011 for all three language pairs (German, Spanish and French). The feature weights for all the systems are tuned using the dev set news-dev2009a. We separately tune the baseline system (cept.500) and the phrase-based system (phrase.200) and do not hold the lambda vector constant like before.

Baseline Systems: We also compared our system with i) Moses (Koehn et al., 2007), ii) Phrasal[14] (Cer et al., 2010), and iii) Ncode (Crego et al., 2011).

We used the default stack sizes of 100 for Moses[15], 200 for Phrasal, 25 for Ncode (with 2^m stacks). A 5-gram English language model is used. Both phrase-based systems use 20-best phrases for translation, Ncode uses 25-best tuple translations. The training and test data for Ncode was tagged using *TreeTagger* (Schmid, 1994). All the baseline systems used lexicalized reordering model. A hard reordering limit[16] of 6 words is used as a default in

[13] We did not include the results of Spanish in the previous section due to space limitations but these are similar to those of the French-to-English translation task.

[14] Phrasal provides two extensions to Moses: i) hierarchical reordering model (Galley and Manning, 2008) and ii) discontinuous phrases (Galley and Manning, 2010).

[15] Using stacks sizes from $200-1000$ did not improve results.

[16] We tried to increase the distortion limit in the baseline sys-

both the baseline phrase-based systems. Amongst the other defaults we retained the hard source gap penalty of 15 and a target gap penalty of 7 in Phrasal. We provide Moses and Ncode with the same post-edited alignments[17] from which we removed target-side discontinuities. We feed the original alignments to Phrasal because of its ability to learn discontinuous source and target phrases. All the systems use MERT for the optimization of the weight vector.

	$\mathbf{M_s}$	$\mathbf{P_d}$	$\mathbf{N_c}$	$\mathbf{C_{500}}$	$\mathbf{P_{200}}$
German-to-English					
MT09	18.73*	19.00*	18.37*	19.06*	19.66
MT10	18.58*	18.96*	18.64*	19.12*	19.70
MT11	17.38*	17.58*	17.49*	17.87*	18.19
French-to-English					
MT09	24.61*	24.73*	24.28*	24.94*	25.27
MT10	23.69*	23.09*	23.96	23.90*	24.25
MT11	25.17*	25.55*	24.92*	25.40*	25.92
Spanish-to-English					
MT09	24.38*	24.63	24.72	24.48*	24.72
MT10	25.55*	25.66*	25.87	25.68*	26.10
MT11	25.72*	26.17*	26.36*	26.48	26.67

Table 2: Comparison on 3-Test Sets – $\mathbf{M_s}$ = Moses, $\mathbf{P_d}$ = Phrasal (Discontinuous Phrases), $\mathbf{N_c}$ = Ncode, $\mathbf{C_{500}}$ = Cept.500, $\mathbf{P_{200}}$ = Phrase.200

Table 2 compares the performance of our phrase-based decoder against the baselines. Our system shows an improvement over all the baseline systems for the G-E pair, in 11 out of 12 cases in the F-E pair and in 8 out of 12 cases in the S-E language pair. We mark a baseline with "*" to indicate that our decoder shows an improvement over this baseline result which is significant at the $p \leq 0.05$ level.

7 Conclusion and Future Work

We proposed a combination of using a model based on minimal units and decoding with phrases. Modeling with minimal units enables us to learn local and non-local dependencies in a unified manner and avoid spurious segmentation ambiguities. However, using minimal units also in the search presents a significant challenge because of the poor translation coverage, inaccurate future-cost estimates and the pruning of the correct hypotheses. Phrase-based SMT on the other hand overcomes these drawbacks by using larger translation chunks during search. However, the drawback of the phrase-based model is the phrasal independence assumption, spurious ambiguity in segmentation and a weak mechanism to handle non-local reorderings. We showed that combining a model based on minimal units with phrase-based decoding can improve both search accuracy and translation quality. We also showed that the phrasal information can be indirectly used in cept-based decoding with improved results. We tested our system against the state-of-the-art phrase-based and N-gram-based systems, for German-to-English, French-to-English, and Spanish-to-English for three standard test sets. Our system showed statistically significant improvements over all the baseline systems in most of the cases. We have shown the benefits of using phrase-based search with a model based on minimal units. In future work, we would like to study whether a phrase-based system like Moses or Phrasal can profit from an OSM-style or N-gram-style feature. Feng et al. (2010) previously showed that adding a linearized source-side language model in a phrase-based system helped. It would also be interesting to study whether the insight of using minimal units for modeling and phrase-based search would hold for hierarchical SMT. Vaswani et al. (2011) recently showed that a Markov model over the derivation history of minimal rules can obtain the same translation quality as using grammars formed with composed rules.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful feedback and suggestions. Nadir Durrani and Alexander Fraser were funded by Deutsche Forschungsgemeinschaft grant Models of Morphosyntax for Statistical Machine Translation. Nadir Durrani was partially funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658. Helmut Schmid was supported by Deutsche Forschungsgemeinschaft grant SFB 732. This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. This publication only reflects the authors' views.

tems to 15 (in G-E experiments) as used in our systems but the results dropped significantly in case of Moses and slightly for Phrasal so we used the default limits for both decoders.

[17]Using post-processed alignments gave slightly better results than the original alignments for these baseline systems. Details are omitted due to space limitation.

References

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and R. L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Daniel Cer, Michel Galley, Daniel Jurafsky, and Christopher D. Manning. 2010. Phrasal: A Statistical Machine Translation Toolkit for Exploring New model Features. In *Proceedings of the NAACL HLT 2010 Demonstration Session*, pages 9–12, Los Angeles, California, June.

Marta R. Costa-jussà, Josep M. Crego, David Vilar, José A.R. Fonollosa, José B. Mariño, and Hermann Ney. 2007. Analysis and System Combination of Phrase- and N-Gram-Based Statistical Machine Translation Systems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 137–140, Rochester, New York, April.

Josep M. Crego and José B. Mariño. 2006. Improving Statistical MT by Coupling Reordering and Decoding. *Machine Translation*, 20(3):199–215.

Josep M. Crego, José B. Mariño, and Adrià de Gispert. 2005. Reordered Search and Unfolding Tuples for N-Gram-Based SMT. In *Proceedings of the 10th Machine Translation Summit (MT Summit X)*, pages 283–289, Phuket, Thailand.

Josep M. Crego, François Yvon, and José B. Mariño. 2011. Ncode: an Open Source Bilingual N-gram SMT Toolkit. *The Prague Bulletin of Mathematical Linguistics*, (96):49–58.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A Joint Sequence Translation Model with Integrated Reordering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1045–1054, Portland, Oregon, USA, June.

Minwei Feng, Arne Mauser, and Hermann Ney. 2010. A Source-side Decoding Sequence Model for Statistical Machine Translation. In *Conference of the Association for Machine Translation in the Americas 2010*, Denver, Colorado, USA, October.

Michel Galley and Christopher D. Manning. 2008. A Simple and Effective Hierarchical Phrase Reordering Model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, Hawaii, October.

Michel Galley and Christopher D. Manning. 2010. Accurate Non-Hierarchical Phrase-Based Translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 966–974, Los Angeles, California, June. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of HLT-NAACL*, pages 127–133, Edmonton, Canada.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *International Workshop on Spoken Language Translation 2005*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007 Demonstrations*, Prague, Czech Republic.

Philipp Koehn. 2004a. Pharaoh: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. In *AMTA*, pages 115–124.

Philipp Koehn. 2004b. Statistical Significance Tests for Machine Translation Evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July.

José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, José A. R. Fonollosa, and Marta R. Costa-jussà. 2006. N-gram-Based Machine Translation. *Computational Linguistics*, 32(4):527–549.

Franz J. Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Franz J. Och and Hermann Ney. 2004. The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(1):417–449.

Franz J. Och, Christoph Tillmann, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 20–28, University of Maryland, College Park, MD.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Morristown, NJ, USA.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Intl. Conf. Spoken Language Processing*, Denver, Colorado.

Christoph Tillmann and Tong Zhang. 2005. A Localized Prediction Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 557–564, Ann Arbor, Michigan, June.

Ashish Vaswani, Haitao Mi, Liang Huang, and David Chiang. 2011. Rule Markov Models for Fast Tree-to-String Translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 856–864, Portland, Oregon, USA, June.

Dekai Wu. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377–403.

Omar F. Zaidan. 2009. Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems. *The Prague Bulletin of Mathematical Linguistics*, 91:79–88.