# Extending a Probabilistic Phrase Alignment Approach for SMT

*Mridul Gupta, Sanjika Hewavitharana and Stephan Vogel*

Language Technologies Institute, Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{mridulg, sanjika, vogel+}@cs.cmu.edu

## Abstract

Phrase alignment is a crucial step in phrase-based statistical machine translation. We explore a way of improving phrase alignment by adding syntactic information in the form of chunks as soft constraints guided by an in-depth and detailed analysis on a hand-aligned data set. We extend a probabilistic phrase alignment model that extracts phrase pairs by optimizing phrase pair boundaries over the sentence pair [1]. The boundaries of the target phrase are chosen such that the overall sentence alignment probability is optimal. Viterbi alignment information is also added in the extended model with a view of improving phrase alignment. We extract phrase pairs using a relatively larger number of features which are discriminatively trained using a large-margin online learning algorithm, i.e., Margin Infused Relaxed Algorithm (MIRA) and integrate it in our approach. Initial experiments show improvements in both phrase alignment and translation quality for Arabic-English on a moderate-size translation task.

## 1. Introduction

Phrase-based statistical machine translation has been around for several years It has been well decribed and discussed in [2] and [3]. Most of these phrase-based approaches rely on robust word alignment strategies for phrase pair extraction like the IBM word alignment models [4]. The now standard approach proposed by [2] relied on heuristics to extract phrase pairs by reading off the Viterbi path generated from word alignment models [5] and using maximum likelihood estimates (MLE) for phrase scoring.

[1] proposed a novel probabilistic phrase extraction algorithm which viewed phrase alignment as a sentence splitting problem (PESA). Given a source phrase, the algorithm finds boundaries of the target phrase by optimizing overall sentence alignment probability. This method does not rely on the traditional Viterbi alignment approach as cited above. Phrase pairs are extracted from a bilingual corpus by searching for sentences pairs, which contain a given source phrase and then finding the optimal target phrase within each sentence pair.

In this paper, we propose to extend this approach by adding more information derived from analysis of syntactic constraints based on chunk information. This information is obtained by parsing the parallel corpus using monolingual shallow parsers on the source and target side. We also add alignment information by reading off the Viterbi alignment path for a sentence pair and comparing the relative position of these alignment points with respect to the rectangular block of the source-target phrase pair, which is described later in the paper. Our proposed approach not only helps in improving the quality of extracted phrase pairs but also improves relative quality of translation against the baseline.

Adding linguistically motivated information in phrase-based SMT systems proves to be a tradeoff between unlinguistically motivated phrase pair extraction from parallel text versus incorporating benefits of linguistic analyses derived before training time. Hard linguistic constraints improve quality somewhat but lose out on coverage. It is thus important to use this linguistic *a priori* knowledge as a 'soft constraint' rather than forcing the MT system to completely ignore unlinguistic but strong mappings in the parallel corpus which could result in deterioration of performance. This fact has been emphasized in previous works of [6], [7] and [8].

As stated above, we extend the translation model by adding a set of features based on syntax and alignment. It is then important to combine features in such a way that the phrase extraction step is optimized over aligning the entire sentence pair. Hence, we use an online large-margin training algorithm, i.e., Margin Infused Relaxed Algorithm, (MIRA) developed by [9] to optimize weights over an extended set of features optimized towards an oracle selection. Online discriminative learning algorithms have been popular in the SMT domain, and researchers have used the MIRA algorithm to train MT systems in the past. For instance, [10], [11] used MIRA algorithm to train MT systems over a large number of features during decoding time. The PESA approach described in [1] only used a manually derived set of weights for optimization, since it relied only on lexical information features obtained from word alignment models.

The main contributions of this work are: **1.** Extending a phrase alignment approach by adding syntax and alignment information. **2.** Incorporating an online large-margin training method to optimize weights during phrase extraction.

In Section 2 we give a brief overview of related work.

Section 3 we describe our analysis based on chunk information. Section 4 describes the baseline phrase extraction system. In sections 5 and 6 we explain our extended approach and training algorithm used during phrase extraction. Section 7 gives our experimental setup and summarizes results. Section 8 concludes the paper and lists some future directions.

## 2. Related work

There has been a strong line of research focused on incorporating syntax in SMT systems, chunk information being one approach. Our work is also based on adding chunk-based information as syntactic features to an existing phrase alignment method. Hence, we focus our attention on incorporating chunk information for SMT.

A chunk has been well defined by [12]. Combining locally grouped words (a constituent) as one translation unit has been shown helpful in improving performance of various machine translation systems. One of the recent work in this line of research is [13]. They built a chunk-based example-based machine translation (EBMT) system in which each chunk is treated as a translation unit. It combined a typical EBMT system and a chunk-based system to produce target translations in the form of linguistically motivated chunks. It backed off to the standard EBMT approach, for translating target fragments that were not chunks. Following the chunk-based paradigm, they adapted standard word alignment models to align chunks by treating each chunk as an individual word. This was done to account for sparseness in terms of statistical evidence for words locally grouped as chunks. They achieved improved performance over baseline systems for Korean-English and Chinese-English translation tasks using this approach.

Some of the previous related work on chunk MT also include that of [14] and [15]. [15] had proposed an SMT approach based on combining chunking knowledge. They decomposed the translation model into three levels: sentence level reordering, chunk mapping and translation of words within a chunk pair. [14] treated each translation unit as a chunk by breaking down the translation model into chunk alignment, and word alignment within chunks for translation. They subsequently performed chunk reordering in a sentence pair chunked on both sides.

Each of the above cited work treats chunks as translation units in the translation model. They either completely back-off to a phrase/word based model in the absence of proper chunks or degrade gracefully in order to produce translations for rest of the sentence. None of these models uses available syntactic information in the form of a soft constraint. However, work proposed by [6] and [11] incorporated syntactic information as soft constraints in a hierarchical phrase-based MT system ([16]) by penalizing phrase pairs that violated these linguistic constraints. The penalty (or, cost) incurred is determined by an optimal combination of feature weights and feature values. They did not altogether ignore or revert to standard phrase-based (hierarchical) models in case of these violations.

We carry out this work in similar vein as that of [6] and [11], wherein chunk-based syntactic restrictions are used as soft constraints. We search for an optimal set of target phrases given a source phrase using this information in the PESA model.

## 3. Chunk-based analysis

We present an analysis for exploring ways to add chunk-based information in the model. This analysis was based on a hand-aligned data set containing 21107 sentence pairs for Arabic-English obtained from the DARPA GALE program. We tried to find automatically generated chunk-to-chunk mappings on the basis of manual word alignment information for the language pair in both directions. Chunk mapping analysis provided important insights with respect to coming up with a set of features for the model. We based our analysis on the following criteria:

- We considered one-to-one chunk alignments based on words within a chunk on the source side aligned only to words within a chunk on the target side and vice versa.
- These include all unaligned words within the chunk pair.
- We also restricted the mappings to have no word alignment links outside of the chunk pairs. This criterion was followed in order to keep chunk alignments restricted, which otherwise could be potentially large.

We included unaligned words in our chunk mappings to incorporate some of the language divergences between the language pair. Our alignment analysis also considered one-to-many chunk mappings looking from both sides. Our analysis based on the above criteria is summarized in figure 1.

First half of Figure 1 outlined in black represents a confusion matrix relating all one-to-one chunk mappings based on the criteria listed above. It can be observed from the figure that some of the most frequently occurring chunks like NP, PP (on the Arabic side) do not have direct one-to-one correspodence with chunk labels like NC, PC on the English side. Although, these labels occur in contiguous sequences on either side and could be seen as translation equivalents of each other for the entire sequence. This leads us to align such sequences as chunk phrase pairs based on IBM1 style alignment as described in section 5.1. However, there are strong correlations between other chunk types like VP – VC and ADJP – ADJC which can be used as high frequency combinations in the model. Cells shaded in grey (in the confusion matrix) show relatively high frequent albeit unexpected chunk mappings whereas those in black show expected chunk mappings. We also see that there is a signicant number of unaligned chunks on both sides (1:0 row/column) in the latter half of the figure which is of concern and something that need to be addressed. The row/column titled "Total" represents the total distribution of each individual chunk label in the hand-aligned corpus. Numbers given in percent-

| | ADJC | ADVC | CONJC | INTJ | NC | PC | PRT | PUNCT | UNK | VC | 1:1 | 1:0 | Total | % |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADJP | 5 | 1 | 0 | 0 | 7 | 1 | 0 | 0 | 0 | 6 | 20 | 8 | 73 | 1.04% |
| ADVP | 0 | 9 | 0 | 0 | 3 | 0 | 0 | 0 | 0 | 1 | 13 | 5 | 28 | 0.40% |
| FRAG | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0.01% |
| INTJ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0.01% |
| LST | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0.01% |
| NAC | 0 | 4 | 1 | 0 | 0 | 2 | 0 | 37 | 136 | 0 | 180 | 100 | 418 | 5.96% |
| NP | 15 | 18 | 0 | 0 | 518 | 223 | 1 | 10 | 3 | 127 | 915 | 359 | 3110 | 44.31% |
| PP | 5 | 12 | 0 | 0 | 89 | 341 | 1 | 3 | 2 | 59 | 512 | 170 | 1326 | 18.89% |
| PUNCT | 0 | 4 | 0 | 0 | 11 | 7 | 0 | 438 | 3 | 7 | 470 | 130 | 625 | 8.90% |
| S | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 6 | 3 | 0 | 12 | 168 | 183 | 2.61% |
| SBAR | 0 | 0 | 0 | 0 | 7 | 52 | 0 | 1 | 0 | 0 | 60 | 43 | 148 | 2.11% |
| SBARQ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.01% |
| UNK | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 12 | 15 | 0.21% |
| VP | 6 | 7 | 0 | 0 | 9 | 2 | 0 | 1 | 4 | 392 | 421 | 148 | 918 | 13.08% |
| WHADVP | 0 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 4 | 8 | 0.11% |
| WHNP | 0 | 0 | 0 | 0 | 61 | 6 | 0 | 0 | 0 | 0 | 67 | 84 | 163 | 2.32% |
| 1:1 | 31 | 62 | 1 | 1 | 705 | 635 | 2 | 497 | 151 | 592 | 2677 | 1232 | 7019 | |
| 1:0 | 4 | 25 | 0 | 0 | 136 | 68 | 6 | 374 | 19 | 105 | 737 | | | |
| Total | 78 | 148 | 5 | 3 | 1720 | 1570 | 25 | 997 | 204 | 1204 | 5954 | | | |
| % | 1.31% | 2.49% | 0.08% | 0.05% | 28.89% | 26.37% | 0.42% | 16.75% | 3.43% | 20.22% | | | | |

Figure 1: *Chunk analysis based on hand aligned Arabic-English parallel corpus. Rows represent chunks on Arabic side and columns represent chunks on English side.*

ages denote relative distribution of chunks in the bilingual corpus.

We also analyzed one-to-many and many-to-many chunk mappings, but due to space constraints, we cannot list them all. An important conclusion from this analysis was to consider aligning these continuous chunk sequences as 'chunk phrase pairs' in the extended model.

## 4. Phrase pair extraction as sentence splitting (PESA)

We describe in brief the method for phrase pair extraction as proposed in [1]. Traditional approaches in the past focused on extracting phrase pairs as a post-processing step of different word alignment methods, most prominently, IBM word alignment models. The post-processing step involves reading off a combination of alignment points on the Viterbi paths generated by running word alignment models in both directions (source to target, target to source) to extract phrase pairs. Some of these combinations over alignment points have been proposed by [5].

The method proposed in [1] optimizes the target phrase boundary ($\tilde{\mathbf{e}}$) given a source phrase ($\tilde{\mathbf{f}}$) in a sentence pair ($\mathbf{f}$, $\mathbf{e}$) from a bilingual parallel corpus. The method constrains the calculation of word alignment (for instance, IBM1 model) for phrase pairs to reduce the potentially large search space for phrase pair alignments. The constraints are defined as follows:

**1.** IBM1 model probabilities are summed for words inside the target phrase for words inside a source phrase for phrase alignment. Similarly, for words that lie outside of the source phrase, probabilities are only summed up for words that correspondingly lie outside of a candidate target phrase within a given sentence pair.

**2.** IBM1 model has a uniform distribution of 1/*I* for position alignment, where *I* is the length of the target sentence. In this case, it is modified to 1/*l* for words inside the source phrase and 1/(*I* - *l*), for words outside the source phrase, where *l* is the length of the target phrase.

Mathematically, the constrained IBM1 style phrase alignment probability is represented as:

$$
\begin{aligned}
p_{i_1,i_2}(f|e) \quad = \quad & \frac{1}{(I-l)} \prod_{j=1}^{j_1-1} \sum_{i \notin (i_1 \ldots i_2)} p(f_j|e_i) \\
& \times \frac{1}{l} \prod_{j=j_1}^{j_2} \sum_{i=i_1}^{i_2} p(f_j|e_i) \\
& \times \frac{1}{(I-l)} \prod_{j=j_2+1}^{J} \sum_{i \notin (i_1 \ldots i_2)} p(f_j|e_i) \quad (1)
\end{aligned}
$$

where, $p_{i_1,i_2}(f|e)$ is the sentence alignment probability in the source to target direction. $j_1$, $j_2$ are the start and end positions of a given source phrase in the source sentence, respectively. $i_1$, $i_2$ are target phrase boundaries, optimized over the sentence pair. Similarly, sentence alignment probability is also calculated for the reverse direction (target to source), $p_{i_1,i_2}(e|f)$.

[1] noted that the probability terms calculated in equation (1) and in the reverse direction may lead to weaker alignment scores within a phrase pair even though the overall sentence alignment score may be good. This is more likely in the case of longer sentence pairs. Hence, they introduced phrase alignment probability scores calculated only within phrase pairs over IBM1 style alignment using probabilities from the trained lexicon. They calculated both raw as well as normalized scores in order to make probabilities sum to one. These scores are calculated as:

$$
p(\tilde{f}|\tilde{e}) = \prod_{j=j_1}^{j_2} \sum_{i=i_1}^{i_2} p(f_j|e_i) \quad (2)
$$

$$p_{norm}(\tilde{f}|\tilde{e}) = \prod_{j=j_1}^{j_2} \sum_{i=i_1}^{i_2} \frac{p(f_j|e_i)}{\sum_{i'=1}^{I} p(f_j|e_{i'})} \quad (3)$$

Equation (2) represents probability alignment score of just a phrase pair as opposed to the entire sentence pair in equation (1). Equation (3) represents renormalized scores over phrase pair alignment. [1] computes the scores for both directions. Hence, a total of six scores (sentence alignment, phrase alignment and renormalized scores, in both directions) are obtained in the PESA system.

These scores are then combined in a weighted log-linear manner as follows:

$$(i_1, i_2) = \underset{(i_1,i_2)}{\operatorname{argmax}} \left\{ \sum_{k_1=1}^{3} \lambda_{k_1} \log(P_{k_1}^{fe}) + \sum_{k_2=1}^{3} \lambda_{k_2} \log(P_{k_2}^{ef}) \right\} \quad (4)$$

where, $P_k^{fe}$ and $P_k^{ef}$ are alignment scores in respective directions and $\lambda's$ are the weights associated with each alignment scores which are optimized.

This approach forms our baseline for results reported in section 7. We compare results of the extended model against this baseline.

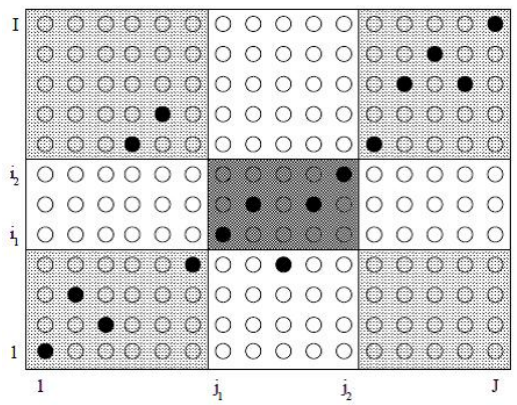


Figure 2: *Phrase pair as sentence splitting (PESA). Black dots indicate alignment points on the Viterbi path. Shaded blocks indicate constrained phrase alignment.*

## 5. Extended model for PESA

In this section we describe an extended model for PESA where we add information based on syntax and alignment in the translation model. Our syntactic approach is based on extracting chunk-based information from the bilingual training corpus.

### 5.1. Chunk-based syntactic constraints

Phrase pairs extracted by the baseline PESA approach are agnostic to syntactic structure. Syntactic structure could play an important role in translation between two relatively distant language pairs as noted by [13], in our case, Arabic and English. The lexical scores for the extracted phrase pairs are computed using IBM4 model trained lexicon. We present a method here to capture chunk-based information available in the form of chunk boundaries and chunk labels.

We use existing statistical monolingual shallow parsers for Arabic and English to produce a chunked bilingual corpus. We used the AMIRA toolkit[1] for Arabic language processing [17], which is a sequence of statistically built segmentation, POS-tagging and base phrase chunking models using support vector machines (SVMs). For English, we used the TreeTagger[2] package developed by [18] for POS-tagging and subsequent base phrase chunking. Some amount of post-processing is required for both sides in order to generate the desired output. For instance, certain tokens are left outside of chunk boundaries by the parsers. We put these left-over tokens within chunk boundaries of a new chunk unit labeled 'UNK'. For more details on chunking strategies please refer [17] and the TreeTagger documentation for Arabic and English, respectively.

We carried out an in-depth analysis in order to determine what all chunk-based information should be captured in the model as presented in section 3. We now describe the set of features that were used to incorporate chunk information in our model.

#### *5.1.1. Chunk boundary (CB) features*

- We look at the number of source and target phrase boundaries ($p$) that match with their corresponding chunk boundaries ($c$) on the left and right sides. Each match produces a constant bonus. This step generates two features, one each for source and target. Formally,

$$\delta(p, c) = \begin{cases} 1 & \text{if } p = c \\ 0 & \text{otherwise} \end{cases}$$

- In a candidate phrase pair with given source phrase, we try to match the corresponding source chunk boundary with target chunk boundary on either sides, i.e. CB-CB (=*match*), CB-noCB (=*no match*), noCB-noCB (=*match*). Here, CB denotes chunk boundary and noCB denotes region within chunk boundaries. Each *match* fires the feature. This step generates two features, one each for left and right sides.

These features introduce a soft bias towards phrases that are full sequences of chunks.

#### *5.1.2. Phrase length to chunk span*

Given a source phrase and its corresponding candidate target phrase, we compute the ratios of the phrase pair with respect to its chunk span for both source and target, thus generating two features. In other words, we look at the "fraction" of the phrase pair that is fully covered by chunks. The intuition behind this feature is that ideally we would want a sequence of chunks to span the entire length of a phrase.

[1] http://nlp.ldeo.columbia.edu/amira/

[2] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

*5.1.3. Chunk balance feature*

- We compute the absolute difference between the number of chunks on the source and target side for a phrase pair. The intuition is that a balanced number of chunks indicates a better phrase pair. It is represented as:

$$d = |n_c^f - m_c^e|$$

where, $n_c^f$ is the number of chunks in the source phrase ($\tilde{f}$) and, $m_c^e$ is the number of chunks in the target phrase ($\tilde{e}$).
- We also look at the ratio of the number of source and target chunks ($n_c^f$, $m_c^e$) in the phrase pair.

*5.1.4. Chunk label features*

We use chunk label mapping information as observed from our analysis on hand-aligned data.

- We replace the sequence of words in a constituent with its chunk label. We then train a lexicon, using these labels, on the chunked corpus. Each underlying sequence of chunk labels within a phrase pair is treated as a 'chunk phrase pair', ($\tilde{f_c}$, $\tilde{e_c}$) with each ($f_j^c$, $e_i^c$) pair representing a chunk to chunk mapping. We then calculate chunk phrase alignment probability using IBM1 style alignment for both directions. Chunk phrase alignment probability can be represented as:

$$p(\tilde{f_c}|\tilde{e_c}) = \prod_{j=j_1}^{j_2} \sum_{i=i_1}^{i_2} p(f_j^c|e_i^c) \tag{5}$$

Similarly, chunk phrase alignment probability can be computed in the reverse direction, $p(\tilde{e_c}|\tilde{f_c})$. These set of features align sequences of chunks on either side to handle one-to-many and many-to-many chunk mappings.
- We also use indicator features for high frequency combinations like VP–VC, ADJP–ADJV, ADVP–ADVC etc. Co-occurrence of these labels can be viewed as a good sign of phrase pair quality.
- Indicator features looking for presence of different types of labels in chunk phrase pairs are also used. These features are represented as follows:

$$\phi_l(f_j^c) = \begin{cases} 1 & \text{if } l = f_j^c \\ 0 & \text{otherwise} \end{cases}$$

for the source phrase and similarly for the target phrase, where $l$ is the chunk label.

### 5.2. Alignment features

Apart from capturing syntactic information explained in section 5.1, we also added alignment features in the extended PESA model. [1] describes a method for finding optimal target phrase boundaries ($i_1$, $i_2$) given the source phrase boundaries ($j_1$, $j_2$), and the sentence pair. As can be seen in figure 2 from [1], alignment is restricted to the grey-colored areas. The black spots indicate alignment points for the sentence pair on the Viterbi path. The rectangular block or the candidate phrase pair (shaded in dark grey) need not always include all target words which are aligned to the source phrase according to this Viterbi path. It can have aligned words from outside the block. We therefore, use this information to look at how many aligned words lie outside this block and penalize such an extracted phrase pair accordingly. The features we use to bring alignment information from the Viterbi path are defined as follows:

**Viterbi alignment points within the phrase pair.** We look at the number of alignment points that lie on the Viterbi path and are within the phrase pair. More number of such alignment points indicates a better phrase pair candidate.

**Unaligned inside words.** Similarly, we look at the number of unaligned words within the phrase pair for both source and target side. Less number of unaligned words is an indicator of better phrase alignment quality.

**Inside-outside alignment.** This feature computes the number of inside words aligned to the outside. We compute this feature for both directions i.e., inside source phrase words aligned with outside target words and vice versa.

**Unaligned words on phrase boundaries.** This feature computes the number of unaligned words at the boundaries of source and target phrases. This feature is particularly important in the case where unaligned words are preferred to lie within the rectangular block, rather than on it.

## 6. Online large-margin discriminative training

Online large-margin training methods have been employed successfully in the past for various structure prediction tasks in natural language processing, such as dependency parsing, chunk labeling and statistical machine translation. Online large-margin algorithms like the margin infused relaxed algorithm (MIRA) developed by [9] have been shown to learn weights of a much larger number of features with greater stability because of its ability to update weights after each training instance based on newly learnt margin constraints.

We used the MIRA algorithm to learn weights for all features in our extended PESA model to produce optimal phrase translations. We describe in brief, how we perform the learning step using MIRA.

### 6.1. MIRA algorithm

MIRA algorithm is defined by an update rule which is subject to max-margin constraints with respect to a loss function computed for the predicted output against the reference. These updates are applied in an online manner, i.e., after seeing each training instance per iteration. The weight update is minimized and the change is kept as low as possible with respect to the current weight vector. Change in the weight vector is subject to the constraint that the margin between the reference/oracle and hypothesis must at least be as large

as the loss value.

Formally, the update rule in MIRA is given by,

$$\underset{w}{\text{argmin}} \quad ||w^{i+1} - w^i||^2 + C \cdot \sum_k \xi_k(y_t, y')$$

$$s.t., \quad s(x_t, y_t) - s(x_t, y_k) + \xi_k(y_t, y') \geq L(y_t, y');$$
$$y' \in best_k(x_t; w^i); \quad \xi_k(y_t, y') \geq 0$$

where, *w* is the weight vector, $y_t$ is the oracle translation closest to the reference, $y'$ is in the *k*-best candidate list, *s* ($x$, $y$) is the scoring function while *L*($y_t$, $y'$) is the computed loss. $\xi_k$ is the slack variable whose value is always non-negative and C is the slack constant used in the objective function that determines how aggressively the weight vector is updated after each instance. The weights obtained after each update are averaged at the end of the training step in order to prevent overfitting.

### 6.2. Loss function and oracle selection

We use two different metrics in computing the loss. The "error" or loss is measured in terms of standard word error rate (WER) normalized against the oracle length. We also compute loss in terms of a modified smoothed-BLEU (mBLEU) version. We present a comparison in error rates for optimizing weights using both mBLEU and WER.

We optimize system performace (or accuracy) towards an oracle target phrase which is selected from a *k*-best list of candidate target phrase translations. Reference target phrases cannot always be reached by the phrase extraction system. This is due to the fact that we try to find translation candidates for phrases extracted from hand-aligned data using training data, which is different from the hand-aligned data. Hence, there is no guarantee that the system would always find reference target phrases in the training data. Thus, we need to select an *m*-best list of target phrase translations closest to one or multiple references from a larger *k*-best list of candidate phrases. In our experiments we set *m*=1 i.e., consider only the first-best oracle phrase.

We consider two different metrics, WER and modified BLEU, in our loss function as stated above. BLEU scoring metric was defined on the document level which considers higher order *n*-gram (typically 4) precision scores for the whole document. Hence, to compute BLEU-like scores at the phrase level we need to do some modifications. We smoothed the original BLEU metric and considered only unigram matches in its computation. Our smoothing technique is defined in a manner similar to that of NIST BLEU evaluation script[3]. The smoothed version of BLEU is computed by adding a partial count of $\left(1/2^k\right)$, for each precision score whose matching *n*-gram count is zero, where *k*=1 for the first '*n*' value for which the *n*-gram match count is zero.

The two different loss functions are given by:

$$L(y_t, y') = WER(y_t, y')$$

for optimization using WER and,

$$L(y_t, y') = 1 - mBLEU(y_t, y')$$

for optimization using modified smoothed-BLEU.

We evaluate the error rates in training and testing for MIRA in terms of normalized Levenshtein distance i.e., WER. This metric of evaluation seems more reasonable than a precision/recall type metric since it is desirable to award partial "credit" for candidate target phrases as the reference is not always reachable in our case.

## 7. Experiments and results

We conducted experiments to test the effectiveness of our approach on both phrase alignment and translation quality for Arabic-English. We present some results from the experiments conducted in this section on a moderate size parallel corpus.

### 7.1. Data and experimental setup

Our data consisted of 641,414 Arabic-English parallel sentences (18.5+19.1 million words) from the news domain obtained from LDC. We trained IBM4 model with MGIZA [19] on the dataset in both directions. Alignment points were refined by using the grow-diag-final heuristic as proposed by [3]. We also ran AMIRA toolkit [17] with ATB segmentation on the Arabic side of the parallel corpus and TreeTagger on English side of the parallel corpus for POS-tagging and chunking. MGIZA was again used to train IBM4 alignment model for the chunked corpus consisting only of chunk labels on both sides. We trained a 5-gram language model on the English Gigaword corpus. We used the Moses toolkit [4] for SMT for decoding.

We also experimented with a 21107-sentence parallel hand-aligned corpus for evaluating phrase alignment quality using MIRA algorithm. We extracted reference target phrases for a given set of source phrases as train/test set from this corpus.

### 7.2. Phrase alignment experiments

We evaluated the efficiency of our extended translation model for phrase alignment using MIRA algorithm for training. We experimented with two metrics (cf. section 5.2) for optimization. We extracted a total of 3645 unique source phrases from the hand-aligned corpus and did a 75-25% split for training/testing (2734/911 phrase pairs). Each source phrase had multiple number of reference translations in our setup. We generate a 10-best target phrase candidate list for each source phrase and evaluate the error against the first best candidate. We ran MIRA for 20 iterations per system. The final averaged weight vector obtained is used to extract phrase tables during the decoding step. Training error rates (in WER) for optimization with MIRA using WER and mBLEU as loss functions on the baseline and extended

[3] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13.pl

[4] http://statmt.org/moses/

Table 1: *Test results for phrase alignment quality using MIRA*

| System | Loss | Error Rate (WER) | Loss | Error Rate (WER) |
|---|---|---|---|---|
| Lexical (Baseline) | WER | 31.40% | mBLEU | 22.41% |
| Baseline+Alignment | WER | 30.26% | mBLEU | 22.23% |
| Baseline+Alignment+Chunks | WER | 29.43% | mBLEU | 21.30% |

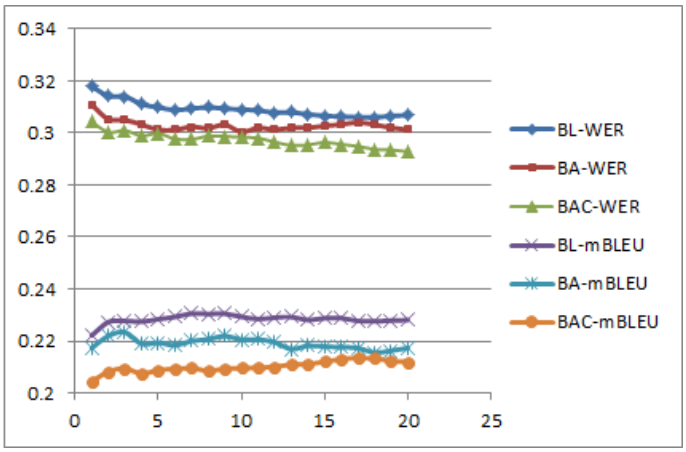


Figure 3: *Training error curves for different models, optimized with two different loss functions. Legend: B-WER (baseline, loss=WER), BA (baseline+alignment features, loss=WER), BAC-WER(baseline+alignment+chunk features, loss=WER). Similarly, for loss=mBLEU. Number of iterations is along $x$-axis and WER along* y*-axis.*

models are shown in figure 3. Test results are shown in table 1. Error rates are measured in WER. We found the value of $C$=0.001 to be optimal in our experiments.

We can see from figure 3 that, MIRA performed much better with mBLEU as the loss function as opposed to WER. We hypothesize the reason for this could be that WER tends to prefer shorter translation candidates, while mBLEU imposes a length penalty that penalizes shorter translations. An analysis on the training set reveals that the average length of the first-best translation candidate for mBLEU is greater than that of WER. There are small but consistent improvements in phrase alignment quality by adding alignment and chunk features to the baseline model. Although, improvement using alignment features only is not significant. There is an improvement of 5% relative to the baseline for optimization with mBLEU and 6% relative for optimization with WER using the full extended model.

### 7.3. Translation experiments

We present initial results for translation experiments on standard NIST MT testsets. We extracted phrase pairs into a phrase table using PESA along with the trained weights. This phrase table was fed in to the Moses decoding pipeline. We did not use the same weights obtained from phrase extraction step in the decoder. This is due to the fact that we have additional features such as language and reordering model scores in the decoder, which requires training to be done afresh when combined with scores from the phrase table. Results shown below are using Moses decoder with a simple distance-based reordering model. We used minimum error rate training [20] to tune weights on MT03 dev set consisting of 663 sentences and evaluated on four NIST eval MT test sets. We report translation results using BLEU and TERp evaluation metrics as shown in table 2. It can be observed that using chunk features and alignment features in different systems, results in initial improvements in both BLEU and TERp scores. There is, on average, an improvement of +1.93 BLEU points on all test sets using chunk features and +1.63 while using alignment features. TERp scores are also lowered for the two systems by -1.14 and -1.01 TERp points on average, respectively. Adding syntax based linguistic information into the system gives better results. Additional improvement is also achieved by adding simplistic features based on Viterbi alignments.

Table 2: *Test results for translation experiments using four standard NIST testsets for Arabic-English. Abbreviations for the systems are as follows: BL: Baseline system using only lexical scores. B+C: System using baseline and chunk features. B+A: System using baseline and alignment features.*

| BLEU Scores | | | | | | |
|---|---|---|---|---|---|---|
| **System** | **MT03 dev** | **MT04** | **MT05** | **MT06** | **MT08** | **Avg** |
| BL | 32.37 | 25.13 | 30.47 | 23.01 | 22.60 | na |
| B+C | 33.97 | **27.05** | 33.01 | **24.92** | 23.96 | **+1.93** |
| B+A | **34.59** | 25.32 | **33.65** | 24.49 | **24.29** | +1.63 |
| TERp Scores | | | | | | |
| **System** | **MT03 dev** | **MT04** | **MT05** | **MT06** | **MT08** | **Avg** |
| BL | 60.58 | 66.69 | 61.01 | **69.32** | 71.73 | na |
| B+C | 60.24 | **64.45** | 59.70 | 69.43 | **70.60** | **-1.14** |
| B+A | **58.81** | 64.72 | **58.77** | 69.51 | 71.71 | -1.01 |

The results reported in this paper are based on one particular set of features in addition to the baseline per system. Hence, we intend to incorporate all sets of features into a single translation model and optimize system performance. This approach of integration of all sources of information in the model requires online discriminative training methods since minimum error rate training algorithm [20] has been shown to be unreliable for a larger number of features. Hence, we would like to integrate the MIRA algorithm into a decoder [21] with which results were first reported for the baseline PESA method in [1]. We also intend to integrate the extended model with an online phrase alignment step ([1]) which eliminates the need for generating large phrase tables offline.

## 8. Conclusion and future work

We presented an approach that can be incorporated successfully into a probabilistic phrase alignment and scoring system. This system treats phrase pair extraction as a sentence splitting problem (PESA) which did not rely on a heuristic-based mechanism to extract and score phrase pairs. The extended model includes syntactic information in the form of chunk-based features. It also includes features based on Viterbi word alignments. The combination of these features is optimized using online margin-based discriminative training methods like MIRA, at the time of phrase extraction which is also an additional contribution to the original approach. The overall approach yielded improvements in phrase alignment and translation quality on Arabic-English translation task.

We intend to integrate MIRA algorithm at decoding time as well to incorporate larger sets of features during decoding and extend it to an online phrase extraction mechanism. We also intend to extend this approach on a full-scale MT system.

## 9. References


[1] Vogel, S., "PESA: Phrase Pair Extraction as Sentence Splitting," In *Proc. MT Summit X*, Phuket, Thailand, September, 2005.

[2] Zens, R., Och, F. J. and Ney, H., "Phrase-Based Statistical Machine Translation," *KI 2002: Advances in Artificial Intelligence, Lecture Notes in Computer Science*, Volume 2479/2002, 35-56, 2002.

[3] Koehn, P., Och, F. J., and Marcu, D., "Statistical Phrase-based Translation," In *Proc. of HLT-NAACL:03*, pages 127–133, Edmonton, Canada, 2003.

[4] Brown, P. F., Pietra S. A., Pietra, V. J., and Mercer, R. L., "The Mathematics of Statistical Machine Translation: Parameter Estimation," *Computational Linguistics*, 19(2):263–311, June, 1993.

[5] Och, F. J., and Ney, H., "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, 29(1):19–51, 2003.

[6] Marton, Y., Resnik, P., "Soft Syntactic Constraints for Hierarchical Phrased-based Translation," In *Proc. ACL-08: HLT*, Columbus, OH, 2008.

[7] Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., and Thayer, I., "Scalable Inference and Training of Context-rich Syntactic Translation Models," In *Proc. of COLING/ACL 2006*, pages 961–968, Sydney, Australia, 2006.

[8] Hassan, H., Sima'an, K., and Way A., "Integrating Supertags into Phrase-based Statistical Machine Translation," In *Proc. of ACL-07*, pages 288–295, Prague, Czech Republic, 2007.

[9] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y., "Online Passive-Aggressive Algorithms," *Journal of Machine Learning Research*, 7:551–585, 2006.

[10] Watanabe, T., Suzuki, J., Tsukada, H., Isozaki, H., "Online Large-Margin Training for Statistical Machine Translation," In *Proc. of EMNLP-CoNLL*, pp 764–773, Prague, Czech Republic, 2007.

[11] Chiang, D., Marton, Y. and Resnik, P., "Online Large-Margin Training of Syntactic and Structural Translation Features," In *Proc. of EMNLP-2008*, Honolulu, Hawaii, 2008.

[12] Abney, S., "Parsing by Chunks," In *Principle- Based Parsing*, pages 257–278. Kluwer Academic Publishers, 1991.

[13] Kim, J. D., Brown, R. D., and Carbonell, J. G., "Chunk-Based EBMT," In *Proc. of 14th Workshop of the European Association for Machine Translation (EAMT-10)*, St. Raphael, France, May, 2010.

[14] Watanabe, T., Sumita, E., and Okuno, H. G., "Chunk-Based Statistical Translation," In *Proc. ACL-2003*, pp 303–310, Sapporo, Japan, 2003.

[15] Koehn, P., and Knight, K., "ChunkMT: Statistical Machine Translation with Richer Linguistic Knowledge," *Unpublished report*, 2002.

[16] Chiang, D., "Hierarchical Phrase-based Translation," *Computational Linguistics*, 33(2):201–228, 2007.

[17] Diab, M., "Second Generation Tools (AMIRA 2.0): Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking," In *Proc. of MEDAR 2nd International Conference on Arabic Language Resources and Tools*, April, 2009.

[18] Schmid, H., "Probabilistic Part-of-Speech Tagging Using Decision Trees," In *Proc. of International Conference on New Methods in Language Processing*, September 1994.

[19] Gao, Q., and Vogel, S., "Parallel Implementations of Word Alignment Tool," In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, 2008.

[20] Och, F. J., "Minimum Error Rate Training in Statistical Machine Translation," In *Proc. of ACL-2003*, pages 160–167, Sapporo, Japan, 2003.

[21] Vogel, S., "SMT Decoder Dissected: Word Reordering," In *Proceedings of Int. Conf. on Natural Language Processing and Knowledge Engineering (NLP-KE)*, Beijing, China, 2003.