Direct Translation Model 2

Abraham Ittycheriah and Salim Roukos
IBM T.J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598
{abei,roukos}@us.ibm.com

Abstract

This paper presents a maximum entropy machine translation system using a minimal set of translation blocks (phrase-pairs). While recent phrase-based statistical machine translation (SMT) systems achieve significant improvement over the original source-channel statistical translation models, they 1) use a *large* inventory of blocks which have significant overlap and 2) limit the use of training to just a few parameters (on the order of ten). In contrast, we show that our proposed minimalist system (DTM2) achieves equal or better performance by 1) recasting the translation problem in the traditional statistical modeling approach using blocks with no overlap and 2) relying on training most system parameters (on the order of millions or larger). The new model is a direct translation model (DTM) formulation which allows easy integration of additional/alternative views of both source and target sentences such as segmentation for a source language such as Arabic, part-of-speech of both source and target, etc. We show improvements over a state-of-the-art phrase-based decoder in Arabic-English translation.

1 Introduction

Statistical machine translation takes a source sequence, $\boldsymbol{S} = [s_1\ s_2\ \ldots\ s_K]$, and generates a target sequence, $\boldsymbol{T}^* = [t_1\ t_2\ \ldots\ t_L]$, by finding the most likely translation given by:

$$\boldsymbol{T}^* = \arg\max_{\boldsymbol{T}} p(\boldsymbol{T}|\boldsymbol{S}).$$

1.1 Block selection

Recent statistical machine translation (SMT) algorithms generate such a translation by incorporating an inventory of bilingual phrases (Och and Ney, 2000). A *m-n* phrase-pair, or block, is a sequence of m source words paired with a sequence of n target words. The inventory of blocks in current systems is highly redundant. We illustrate the redundancy using the example in Table 1 which shows a set of phrases that cover the two-word Arabic fragment "*lljnp Almrkzyp*" whose alignment and translation is shown in Figure 1. One notices the significant overlap between the various blocks including the fact the output target sequence "*of the central committee*" can be produced in at least two different ways: 1) as 2-4 block "*lljnp Almrkzyp | of the central committee*" covering the two Arabic words, or 2) by using the 1-3 block "*Almrkzyp | of the central*" followed by covering the first Arabic word with the 1-1 block "*lljnp | committee*". In addition, if one adds one more word to the Arabic fragment in the third position such as the block "*AlSyny | chinese*" the overlap increases significantly and more alternate possibilities are available to produce an output such as the "*of the central chinese committee.*"

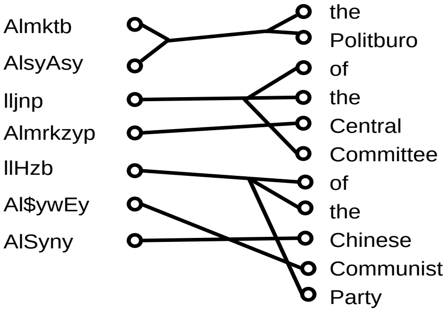

Figure 1: Example of Arabic snipet and alignment to its English translation.

In this work, we propose to only use 1-n blocks and avoid completely the redundancy obtained by the use of m-n blocks for $m > 1$ in current phrase-based systems. We discuss later how by defining appropriate features in the translation model, we capture the important dependencies required for producing n-long fragments for an m-word input sequence including the reordering required to produce more fluent output. So in Table 1 only the blocks corresponding to a single Arabic word are in the block inventory. To differentiate this work from previous approaches in

Proceedings of NAACL HLT 2007, pages 57–64,
Rochester, NY, April 2007. ©2007 Association for Computational Linguistics

lljnp	**Almrkzyp**
committee	central
of the commission	the central
commission	of the central
of the committee	of central
the committee	and the central
of the commission on	and central
the commission	, central
committee of	's central
...	...
of the central committee(11)	
of the central committee of (11)	
the central committee of (8)	
central committee(7)	
committee central (2)	
central committee , (2)	
...	

Table 1: Example Arabic-English blocks showing possible 1-n and 2-n blocks ranked by frequency. Block count is given in () for 2-n blocks.

direct modeling for machine translation, we call our current approach DTM2 (Direct Translation Model 2).

1.2 Statistical modeling for translation

Earlier work in statistical machine translation (Brown et al., 1993) is based on the "noisy-channel" formulation where

$$\boldsymbol{T}^* = \arg\max_{\boldsymbol{T}} p(\boldsymbol{T}|\boldsymbol{S}) = \arg\max_{\boldsymbol{T}} p(\boldsymbol{T})p(\boldsymbol{S}|\boldsymbol{T}) \quad (1)$$

where the target language model $p(\boldsymbol{T})$ is further decomposed as

$$p(\boldsymbol{T}) \propto \prod_i p(t_i|t_{i-1},\ldots,t_{i-k+1})$$

where k is the order of the language model and the translation model $p(\boldsymbol{S}|\boldsymbol{T})$ has been modeled by a sequence of five models with increasing complexity (Brown et al., 1993). The parameters of each of the two components are estimated using Maximum Likelihood Estimation (MLE). The LM is estimated by counting n-grams and using smoothing techniques. The translation model is estimated via the EM algorithm or approximations that are bootstrapped from the previous model in the sequence as introduced in (Brown et al., 1993). As is well known, improved results are achieved by modifying the Bayes factorization in Equation 1 above by weighing each distribution differently as in:

$$p(\boldsymbol{T}|S) \propto p^{\alpha}(\boldsymbol{T})p^{1-\alpha}(\boldsymbol{S}|\boldsymbol{T}) \quad (2)$$

This is the simplest MaxEnt[1] model that uses two feature functions. The parameter α is tuned on a development set (usually to improve an error metric instead of MLE). This model is a special case of the Direct Translation Model proposed in (Papineni et al., 1997; Papineni et al., 1998) for language understanding; (Foster, 2000) demostrated perplexity reductions by using direct models; and (Och and Ney, 2002) employed it very successfully for language translation by using about ten feature functions:

$$p(\boldsymbol{T}|\boldsymbol{S}) = \frac{1}{Z}\exp\sum_i \lambda_i\phi_i(\boldsymbol{S},\boldsymbol{T})$$

Many of the feature functions used for translation are MLE models (or smoothed variants). For example, if one uses $\phi_1 = log(p(\boldsymbol{T}))$ and $\phi_2 = log(p(\boldsymbol{S}|\boldsymbol{T}))$ we get the model described in Equation 2. Most phrase-based systems, including the baseline decoder used in this work use feature functions:

- a target word n-gram model (e.g., $n = 5$),
- a target part-of-speech n-gram model ($n \geq 5$),
- various translation models such as a block inventory with the following three varieties: 1) the unigram block count, 2) a model 1 score $p(\boldsymbol{s}_i|\boldsymbol{t}_i)$ on the phrase-pair, and 3)a model 1 score for the other direction $p(\boldsymbol{t}_i|\boldsymbol{s}_i)$,
- a target word count penalty feature $|\boldsymbol{T}|$,
- a phrase count feature,
- a distortion model (Al-Onaizan and Papineni, 2006).

The weight vector $\boldsymbol{\lambda}$ is estimated by tuning on a rather *small* (as compared to the training set used to define the feature functions) development set using the BLEU metric (or other translation error metrics). Unlike MaxEnt training, the method (Och, 2003) used for estimating the weight vector for BLEU maximization are not computationally scalable for a large number of feature functions.

2 Related Work

Most recent state-of-the-art machine translation decoders have the following aspects that we improve upon in this work: 1) block style, and 2) model parameterization and parameter estimation. We discuss each item next.

[1] The subfields of log-linear models, exponential family, and MaxEnt describe the equivalent techniques from different perspectives.

2.1 Block style

In order to extract phrases from alignments available in one or both directions, most SMT approaches use a heuristic such as *union*, *intersection*, *inverse projection constraint*, etc. As discussed earlier, these approaches result in a large overlap between the extracted blocks (longer blocks overlap with all the shorter subcomponents blocks). Also, slightly restating the advantages of phrase-pairs identified in (Quirk and Menezes, 2006), these blocks are effective at capturing context including the encoding of non-compositional phrase pairs, and capturing local reordering, but they lack variables (e.g. embedding between $ne \ldots pas$ in French), have sparsity problems, and lack a strategy for global reordering. More recently, (Chiang, 2005) extended phrase-pairs (or blocks) to hierarchical phrase-pairs where a grammar with a single non-terminal allows the embedding of phrases-pairs, to allow for arbitrary embedding and capture global reordering though this approach still has the high overlap problem. However, in (Quirk and Menezes, 2006), the authors investigate minimum translation units (MTU) which is a refinement over a similar approach by (Banchs et al., 2005) to eliminate the overlap issue. The MTU approach picks all the minimal blocks subject to the condition that no word alignment link crosses distinct blocks. They do not have the notion of a block with a variable (a special case of the hierarchical phrase-pairs) that we employ in this work. They also have a weakness in the parameter estimation method; they rely on an n-gram language model on blocks which inherently requires a large bilingual training data set.

2.2 Estimating Model Parameters

Most recent SMT systems use blocks (i.e. phrase-pairs) with a few real valued "informative" features which can be viewed as an indicator of how probable the current translation is. As discussed in Section 1.2, these features are typically MLE models (e.g. block translation, Model 1, language model, etc.) whose scores are log-linearly combined using a weight vector, λ_f where f is a particular feature. The λ_f are trained using a held-out corpus using maximum BLEU training (Och, 2003). This method is only practical for a small number of features; typically, the number of features is on the order of 10 to 20.

Recently, there have been several discriminative approaches at training large parameter sets including (Tillmann and Zhang, 2006) and (Liang et al., 2006). In (Tillmann and Zhang, 2006) the model is optimized to produce a block orientation and the target sentence is used only for computing a sentence level BLEU. (Liang et al., 2006) demonstrates a discriminatively trained system for machine translation that has the following characteristics: 1) requires a varying update strategy (local vs. bold) depending on whether the reference sentence is "reachable" or not, 2) uses sentence level BLEU as a criterion for selecting which output to update towards, and 3) only trains on limited length (5-15 words) sentences.

So both methods fundamentally rely on a prior decoder to produce an "N-best" list that is used to find a target (using max BLEU) for the training algorithm. The methods to produce an "N-best" list tend to be not very effective since most alternative translations are minor differences from the highest scoring translation and do not typically include the reference translation (particularly when the system makes a large error).

In this paper, the algorithm trains on all sentences in the test-specific corpus and crucially, the algorithm directly uses the target translation to update the model parameters. This latter point is a critical difference that contrasts to the major weakness of the work of (Liang et al., 2006) which uses a top-N list of translations to select the maximum BLEU sentence as a target for training (so called local update).

3 A Categorization of Block Styles

In (Brown et al., 1993), multi-word "cepts" (which are realized in our block concept) are discussed and the authors state that when a target sequence is sufficiently different from a word by word translation, only then should the target sequence should be promoted to a cept. This is in direct opposition to phrase-based decoders which utilize all possible phrase-pairs and limit the number of phrases only due to practical considerations. Following the perspective of (Brown et al., 1993), a minimal set of phrase blocks with lengths (m, n) where either m or n must be greater than zero results in the following types of blocks:

1. $n = 0$, source word producing nothing in the target language (deletion block),
2. $m = 0$, spontaneous target word (insertion block),
3. $m = 1$ and $n \geq 1$, a source word producing n target words including the possibility of a variable (denoted by $\mathbf{X}$) which is to be filled with other blocks from the sentence (the latter case called a discontiguous block)
4. $m \geq 1$ and $n = 1$, a sequence of source words producing a single target words including the possibility of a variable on the source side (as in the French ne...pas translating into not, called multi-word singletons) in the source sequence

5. $m > 1$ and $n > 1$, a non-compositional phrase translation

In this paper, we restrict the blocks to Types 1 and 3. From the example in Figure 1, the following blocks are extracted:

- lljnp ⇒ of the **X** Committee
- Almrkzyp ⇒ Central
- llHzb ⇒ of the **X** Party
- Al$ywEy ⇒ Communist
- AlSyny ⇒ Chinese.

These blocks can now be considered more "general" and can be used to generate more phrases compared to the blocks shown in Table 1. These blocks when utilized independently of the remainder of the model perform very poorly as all the advantages of blocks are absent. These advantages are obtained using the features to be described below. Also, we store with a block additional information such as: (a) alignment information, and (b) source and target analysis. The target analysis includes part of speech and for each target string a list of part of speech sequences are stored along with their corpus frequencies.

The first alignment shown in Figure 1 is an example of a Type 5 non-compositional block; although this is not currently addressed by the decoder, we plan to handle such blocks in the future.

4 Algorithm

A classification problem can be considered as a mapping from a set of histories, $\mathcal{S}$, into a set of futures, $\mathcal{T}$. Traditional classification problems deal with a small finite set of futures usually no more than a few thousands of classes.

Machine translation can be cast into the same framework with a much larger future space. In contrast to the current global models, we decompose the process into a sequence of steps. The process begins at the left edge of a sentence and for practical reasons considers a window of source words that could be translated. The first action is to jump a distance, j to a source position and to produce a target string, $\boldsymbol{t}$ corresponding to the source word at that position. The process then marks the source position as having been visited and iterates till all source words have been visited. The only wrinkle in this relatively simple process is the presence of a variable in the target sequence. In the case of a variable, the source position is marked as having been partially visited. When a partially visited source position is visited again, the target string to the right of the variable is output and the process is iterated. The distortion or jump from the previously translated source word, j in training can vary widely due to automatic sentence alignment that is used to create the parallel corpus. To limit the sparseness created by these longer jumps we cap the jump to a window of source words (-5 to 5 words) around the last translated source word; jumps outside the window are treated as being to the edge of the window.

We combine the above translation model with a n-gram language model as in

$$\begin{aligned} p(T, j|S) &= \prod_i p(t_i, j|s_i) \\ &\approx \prod_i \lambda_{\text{LM}} p(t_i|t_{i-1}, \ldots, t_{i-n}) + \\ &\qquad \lambda_{\text{TM}} p(t_i, j|s_i) \end{aligned}$$

This mixing allows the use of language model built from a very large monolingual corpus to be used with a translation model which is built from a smaller parallel corpus. In the rest of this paper, we are concerned only with the translation model.

The minimum requirements for the algorithm are (a) parallel corpus of source and target languages and (b) word-alignments. While one can use the EM algorithm to train this hidden alignment model (the jump step), we use Viterbi training, i.e. we use the most likely alignment between target and source words in the training corpus to estimate this model. We assume that each sentence pair in the training corpus is word-aligned (e.g. using a MaxEnt aligner (Ittycheriah and Roukos, 2005) or an HMM aligner (Ge, 2004)). The algorithm performs the following steps in order to train the maximum entropy model: (a) block extraction, (b) feature extraction, and (c) parameter estimation. Each of the first two steps requires a pass over the training data and parameter estimation requires typically 5-10 passes over the data. (Della Pietra et al., 1995) documents the Improved Iterative Scaling (IIS) algorithm for training maximum entropy models. When the system is restricted to 1-N type blocks, the future space includes all the source word positions that are within the skip window and all their corresponding blocks. The training algorithm at the parameter estimation step can be concisely stated as:

1. For each sentence pair in the parallel corpus, walk the alignment in source word order.

2. At each source word, the alignment identifies the "true" block.

3. Form a window of source words and allow all blocks at source words to generate at this generation point.

4. Apply the features relevant to each block and compute the probability of each block.

5. Form the MaxEnt polynomials(Della Pietra et al., 1995) and solve to find the update for each feature.

We will next discuss the prior distribution used in the maximum entropy model, the block extraction method and the feature generation method and discuss differences with a standard phrase based decoder.

4.1 Prior Distribution

Maximum entropy models are of the form,

$$p(\boldsymbol{t}, j|\boldsymbol{s}) = \frac{p_0(\boldsymbol{t}, j|\boldsymbol{s})}{Z} \exp \sum_i \lambda_i \phi_i(\boldsymbol{t}, j, \boldsymbol{s})$$

where p_0 is a prior distribution, Z is a normalizing term, and $\phi_i(\boldsymbol{t}, j, \boldsymbol{s})$ are the features of the model. The prior distribution can contain any information we know about our future and in this work we utilize the normalized phrase count as our prior. Strictly, the prior has to be uniform on the set of futures to be a "maximum" entropy algorithm and choices of other priors result in minimum divergence models. We refer to both as a maximum entropy models.

The practical benefit of using normalized phrase count as the prior distribution is for rare translations of a common source words. Such a translation block may not have a feature due to restrictions in the number of features in the model. Utilizing the normalized phrase count prior, the model is still able to penalize such translations. In the best case, a feature is present in the model and the model has the freedom to either boost the translation probability or to further reduce the prior.

4.2 Block Extraction

Similar to phrase decoders, a single pass is made through the parallel corpus and for each source word, the target sequence derived from the alignments is extracted. The 'Inverse Projection Constraint', which requires that the target sequence be aligned only to the source word or phrase in question, is then checked to ensure that the phrase pair is consistent. A slight relaxation is made to the traditional target sequence in that variables are allowed if the length of their span is 3 words or less. The length restriction is imposed to reduce the effect of alignment errors. An example of blocks extracted for the romanized arabic words 'lljnp' and 'Almrkzyp' are shown Figure 2, where on the left side are shown the unsegmented Arabic words, the segmented Arabic stream and the corresponding Arabic part-of-speech. On the right, the target sequences are shown with the most frequently occuring part-of-speech and the corpus count of this block.

The extracted blocks are pruned in order to minimize alignment problems as well as optimize the speed during decoding. Blocks are pruned if their corpus count is a factor of 30 times smaller than the most frequent target sequence for the same source word. This results in about 1.6 million blocks from an original size of 3.2 million blocks (note this is much smaller than the 50 million blocks or so that are derived in current phrase-based systems).

4.3 Features

The features investigated in this work are binary questions about the lexical context both in the source and target streams. These features can be classified into the following categories: (a) block internal features, and (b) block context features. Features can be designed that are specific to a block. Such features are modeling the unigram phrase count of the block, which is information already present in the prior distribution as discussed above. Features which are less specific are tied across many translations of the word. For example in Figure 2, the primary translation for 'lljnp' is 'committee' and occurs 920 times across all blocks extracted from the corpus; the final block shown which is 'of the **X** committee' occurs only 37 times but employs a lexical feature 'lljnp committee' which fires 920 times.

4.3.1 Lexical Features

Lexical features are block internal features which examine a source word, a target word and the jump from the previously translated source word. As discussed above, these are shared across blocks.

4.3.2 Lexical Context Features

Context features encode the context surrounding a block by examining the previous and next source word and the previous two target words. Unlike a traditional phrase pair, which encodes all the information lexically, in this approach we define in Table 2, individual feature types to examine a portion of the context. One or more of these features may apply in each instance where a block is relevant. The previous source word is defined as the previously translated source word, but the next source word is always the next word in the source string. At training time, the previously translated source word is found by finding the previous target word and utilizing the alignment to find the previous source word. If the previous target word is unaligned, no context feature is applied.

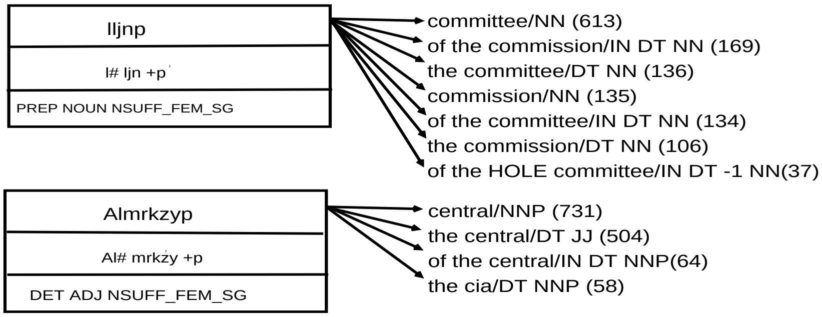
lljnp
l# ljn +p
PREP NOUN NSUFF_FEM_SG
committee/NN (613)
of the commission/IN DT NN (169)
the committee/DT NN (136)
commission/NN (135)
of the committee/IN DT NN (134)
the commission/DT NN (106)
of the HOLE committee/IN DT -1 NN(37)
Almrkzyp
Al# mrkży +p
DET ADJ NSUFF_FEM_SG
central/NNP (731)
the central/DT JJ (504)
of the central/IN DT NNP(64)
the cia/DT NNP (58)

Figure 2: Extracted blocks for 'lljnp' and 'Almrkzyp'.

Feature Name	Feature variables
SRC_LEFT	source left, source word, target word
SRC_RIGHT	source right, source word, target word
SRC_TGT_LEFT	source left, target left, source word, target word
SRC_TGT_LEFT_2	source left, target left, target left 2, source word, target word

Table 2: Context Feature Types

4.3.3 Arabic Segmentation Features

An Arabic segmenter produces morphemes; in Arabic, prefixes and suffixes are used as prepositions, pronouns, gender and case markers. This produces a segmentation view of the arabic source words (Lee et al., 2003). The features used in the model are formed from the Cartesian product of all segmentation tokens with the English target sequence produced by this source word or words. However, prefixes and suffixes which are specific in translation are limited to their English translations. For example the prefix 'Al#' is only allowed to participate in a feature with the English word 'the' and similarly 'the' is not allowed to participate in a feature with the stem of the Arabic word. These restrictions limit the number of features and also reduce the over fitting by the model.

4.3.4 Part-of-speech Features

Part-of-speech taggers were run on each language: the English part of speech tagger is a MaxEnt tagger built on the WSJ corpus and on the WSJ test set achieves an accuracy of 96.8%; the Arabic part of speech tagger is a similar tagger built on the Arabic tree bank and achieves an accuracy of 95.7% on automatically segmented data. The part of speech feature type examines the source and target as well as the previous target and the corresponding previous source part of speech. A separate feature type examines the part of speech of the next source word when the target sequence has a variable.

4.3.5 Coverage Features

These features examine the coverage status of the source word to the left and the source word to the right. During training, the coverage is determined by examining the alignments; the source word to the left is uncovered if its target sequence is to the right of the current target sequence. Since the model employs binary questions and predominantly the source word to the left is already covered and the right source word is uncovered, these features fire only if the left is open or if the right is closed in order to minimize the number of features in the model.

5 Translation Decoder

A beam search decoder similar to phrase-based systems (Tillmann and Ney, 2003) is used to translate the Arabic sentence into English. These decoders have two parameters that control their search strategy: (a) the skip length (how many positions are allowed to be untranslated) and (b) the window width, which controls how many words are allowed to be considered for translation. Since the majority of the blocks employed in this work do not encode local reordering explicitly, the current DTM2 decoder uses a large skip (4 source words for Arabic) and tries all possible reorderings. The primary difference between a DTM2 decoder and standard phrase based decoders is that the maximum entropy model provides a cost estimate of producing this translation using the features described in previous sections. Another difference is that the DTM2 decoder handles blocks with variables. When such a block is proposed, the initial target sequence is first output and the source word position is marked as being partially visited and an index into which segment was generated is kept for completing the visit at a later time. Subsequent extensions of this path can either complete this visit or visit other source words. On a search path, we make a further assumption that only one source position can be in a partially visited state at any point. This greatly reduces the search task and suffices to handle the type of blocks encountered in Arabic to English translation.

6 Experiments

The UN parallel corpus and the LDC news corpora released as training data for the NIST MT06 evaluation are used for all evaluations presented in this paper. A variety of test corpora are now available and we use MT03 as development test data, and test results are presented on MT05. Results obtained on MT06 are from a blind evaluation. For Arabic-English, the NIST MT06 training data contains 3.7M sentence pairs from the UN from 1993-2002 and 100K sentences pairs from news sources. This represents the universe of training data, but for each test set we sample this corpus to train efficiently while also observing slight gains in performance. The training universe is time sorted and the most recent corpora are sampled first. Then for a given test set, we obtain the first 20 instances of n-grams from the test that occur in the training universe and the resulting sampled sentences then form the training sample. The contribution of the sampling technique is to produce a smaller training corpus which reduces the computational load; however, the sampling of the universe of sentences can be viewed as test set domain adaptation which improves performance and is not strictly done due to computational limitations[2]. The 5-gram language model is trained from the English Gigaword corpus and the English portion of the parallel corpus used in the translation model training.

The baseline decoder is a phrase-based decoder that employs *n-m* blocks and uses the same test set specific training corpus described above.

6.1 Feature Type Experiments

There are 15 individual feature types utilized in the system, but in order to be brief we present the results by feature groups (see Table 3): (a) lexical, (b) lexical context, (c) segmentation, (d) part-of-speech, and (e) coverage features. The results show improvements with the addition of each feature set, but the part-of-speech features and coverage features are not statistically significant improvements. The more complex features based on Arabic segmentation and English part-of-speech yield a small improvement of 0.5 BLEU points over the model with only lexical context.

Verb Placement	3
Missing Word	5
Extra Word	5
Word Choice	26
Word Order	3
Other error	1
Total	43

Table 4: Errors on last 25 sentences of MT-03.

7 Error Analysis and Discussion

We analyzed the errors in the last 25 sentences of the MT-03 development data using the broad categories shown in Table 4. These error types are not independent of each other; indeed, incorrect verb placement is just a special case of the word order error type but for this error analysis for each error we take the first category available in this list. Word choice errors can be a result of (a) rare words with few, or incorrect, or no translation blocks (4 times) or (b) model weakness[3] (22 times). In order to address the model weakness type of errors, we plan on investigating feature selection using a language model prior. As an example, consider an arabic word which produces both 'the' (due to alignment errors) and 'the conduct'. An n-gram LM has very low cost for the word 'the' but a rather high cost for content words such as 'conduct'. Incorporating the LM model as a prior should help the maximum entropy model focus its weighting on the content word to overcome the prior information.

8 Conclusion and Future Work

We have presented a complete direct translation model with training of millions of parameters based on a set of minimalist blocks and demonstrated the ability to retain good performance relative to phrase based decoders. Tied features minimize the number of parameters and help avoid the sparsity problems associated with phrase based decoders. Utilizing language analysis of both the source and target languages adds 0.8 BLEU points on MT-03, and 0.4 BLEU points on MT-05. The DTM2 decoder achieved a 1.7 BLEU point improvement over the phrase based decoder on MT-06. In this work, we have restricted the block types to only single source word blocks. Many city names and dates in Arabic can not be handled by such blocks and in future work we intend to investigate the utilization of more complex blocks as necessary. Also, the DTM2 decoder utilized the LM component independently of

[2] Recent results indicate that test set adaptation by test set sampling of the training corpus achieves a cased Bleu of 53.26 on MT03 whereas a general system trained on all data achieves only 51.02

[3] The word occurred with the correct translation in the phrase library with a count more than 10 and yet the system used an incorrect translation.

Feature Types		# of feats (MT03)	MT-03	MT-05	MT-06
Training Size Num. of Sentences			197K	267K	279K
Phrase-based Decoder			51.20	49.06	36.92
DTM2 Decoder Lex Feats	a	439,582	49.70	48.37	
+Lex Context	b	2,455,394	50.45	49.61	
+Seg Feats	c	2,563,338	50.97	49.96	
+POS Feats	d	2,608,352	51.27	49.93	
+Cov Feats	e	2,783,813	51.19	50.00	38.61

Table 3: Bleu scores on MT03-MT06.

the translation model; however, in future work we intend to investigate feature selection using the language model as a prior which should result in much smaller systems.

9 Acknowledgements

This work was partially supported by the Department of the Interior, National Business Center under contract No. NBCH2030001 and Defense Advanced Research Projects Agency under contract No. HR0011-06-2-0001. The views and findings contained in this material are those of the authors and do not necessarily reflect the position or policy of the U.S. government and no official endorsement should be inferred. This paper owes much to the collaboration of the Statistical MT group at IBM.

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 529–536, Sydney, Australia.

Rafael Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, and José B. Marino. 2005. Statistical machine translation of euparl data by using bilingual n-grams. In *Proc. of the ACL Workshop on Building and Using Parallel Texts*, pages 133–136, Ann Arbor, Michigan, USA.

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

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 263–270, Ann Arbor, Michigan, June.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. 1995. Inducing features of random fields. *Technical Report, Department of Computer Science, Carnegie-Mellon University, CMU-CS-95-144*.

George Foster. 2000. A maximum entropy/minimum divergence translation model. In *38th Annual Meeting of the ACL*, pages 45–52, Hong Kong.

Niyu Ge. 2004. Improvement in Word Alignments. *Presentation given at DARPA/TIDES MT workshop.*

Abraham Ittycheriah and Salim Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *HLT '05: Proceedings of the HLT and EMNLP*, pages 89–96.

Young-Suk Lee, Kishore Papineni, and Salim Roukos. 2003. Language model based arabic word segmentation. In *41st Annual Meeting of the ACL*, pages 399–406, Sapporo, Japan.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 761–768, Sydney, Australia.

Franz Josef Och and Hermann Ney. 2000. Statistical machine translation. In *EAMT Workshop*, pages 39–46, Ljubljana, Slovenia.

Franz-Josef Och and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translations. In *40th Annual Meeting of the ACL*, pages 295–302, Philadelphia, PA, July.

Franz Josef Och. 2003. Minimum error rate training in Statistical Machine Translation. In *41st Annual Meeting of the ACL*, pages 160–167, Sapporo, Japan.

Kishore Papineni, Salim Roukos, and R. T. Ward. 1997. Feature-based language understanding. In *EUROSPEECH*, pages 1435–1438, Rhodes,Greece.

Kishore Papineni, Salim Roukos, and R. T. Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *International Conf. on Acoustics, Speech and Signal Processing*, pages 189–192, Seattle, WA.

Chris Quirk and Arul Menezes. 2006. Do we need phrases? challenging the conventional wisdom in statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 9–16, New York, NY, USA.

Christoph Tillmann and Hermann Ney. 2003. Word reordering and a dynamic programming beam search algorithm for Statistical Machine Translation. 29(1):97–133.

Christoph Tillmann and Tong Zhang. 2006. A discriminative global training algorithm for statistical mt. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 721–728, Sydney, Australia.