Dynamic Distortion in a Discriminative Reordering Model for Statistical Machine Translation

Sirvan Yahyaei

School of Electronic Engineering and Computer Science
Queen Mary, University of London
London E1 4NS, UK
sirvan@eecs.qmul.ac.uk

Christof Monz

ISLA, Informatics Institute
University of Amsterdam, Science Park 107
1098 XG Amsterdam, The Netherlands
c.monz@uva.nl

Abstract

Most phrase-based statistical machine translation systems use a so-called distortion limit to keep the size of the search space manageable. In addition, a distance-based distortion penalty is used as a feature to keep the decoder to translate monotonically unless there is sufficient support for a jump from other features, particularly the language models.

To overcome the issue of setting the optimum distortion parameters in the phrase-based decoders and the fact that different sentences have different reordering requirements, a method to predict the necessary distortion limit for each sentence and each hypothesis expansion is proposed. A discriminative reordering model is built for that purpose and also integrated into the decoder as an extra feature. Many lexicalised and syntactic features of the source sentences are employed to predict the next reordering move of the decoder. The model scores each reordering before the sentence translation, so the optimum distortion limit can be estimated based on these score. Various experiments on Turkish to English and Arabic to English pairs are performed and substantial improvements are reported.

1. Introduction

Non-hierarchical phrase-based statistical machine translation systems are one of the most successful machine translation approaches currently available. Various researchers worked on different aspects of these systems and many alternatives and improvements have been proposed. Some of the advantages of the phrase-based systems are fast decoding and better coverage of huge numbers of syntactic and non-syntactic phrases over other approaches [1]. On the other hand, hierarchical and syntax-based methods learn complex reordering rules as a part of the translation model building. There are well-established reordering models to compensate for this in phrase-based systems, however, they are still limited by some parameters to make the search process feasible.

An important parameter in most phrase-based systems, which controls the size of the search space explored by the decoder, is the so-called *distortion limit*. The distortion limit specifies the size of the window which the decoder considers to choose the next source phrase. The best value for this parameter is different for different language pairs. Language pairs such as French and English do not need a long distortion span, since they are very similar in their word order differences and most of the reorderings can be captured by the extracted phrases from the bi-text. On the other side, there are language pairs such as Turkish and English with fundamentally different word orders. Turkish is generally a Subject-Object-Verb (SOV) language, which means for many of the sentences a long reordering is required to translate the verb in the right place in the English sentence. However, with a very rich morphology, Turkish word order can vary and some sentences may not need such long distance reorderings.

To improve the phrase-based systems' reordering capabilities, we aim to build a model that scores different reordering decisions based on lexicalised and syntactic features. In addition, we use this model to guide the decoder to dynamically change the size of the reordering window according to the state of translation. Consider the example sentence in figure 1. We want a model to encourage the decoder to skip the first word (mn), but translate the next four words monotonically (mkAn fY AlYAbAn Ant) and finally jump back to translate the uncovered first word. Thus, we condition our jumps not only on the start and end of the jump, but also on the words jumped over. Additionally, in order to increase the size of the reordering window, we dynamically adjust the distortion limit according the requirements of the reordering model. In other words, the size of the window for hypothesis expansion in the decoder is determined by the current state of the decoder. The latest translated phrase and all the phrases that are about to be translated are taken into account to find the required distortion limit for the next step.

The rest of the paper organised as follows: Section 2 overviews some of the approaches for reordering in phrase-based models. Section 3 investigates the importance of distortion in translation quality and speed. Sections 4 and 5, explain our approach to deal with reordering and Section 6 reports experiments done based on the proposed models. Section 7, concludes the paper.

Proceedings of the 7th International Workshop on Spoken Language Translation
Paris, December 2nd and 3rd, 2010

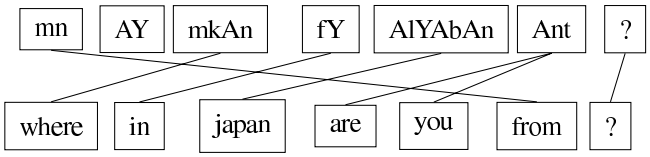

Figure 1: A word alignment example of an Arabic to English sentence pair. The Arabic sentence is romanised according to Buckwalter's method.

2. Related Work

Word order difference between natural languages has been a major challenge in machine translation. Many approaches and models have been proposed to deal with the problem. Syntax-based approaches rely on their syntactic rules to perform the reorderings and produce grammatically correct output. On the other hand, phrase-based approaches deal with most of the local reorderings with the help of extracted phrases and rely on additional features or pre-processing steps to tackle the rest of the reordering requirements.

Transforming the source sentence to comply with the structure of the target language is a method that has been approached from different angles. [2] automatically extracted syntactic rules to transform the source sentence and monotonically decode the transformed source sentence. [3, 4, 5] used a set of hand-crafted rules and syntactic trees to reorder the source sentence. Source reordering has the advantage of being able to employ complex syntactic information specific to the languages involved. Additionally, since reordering the source is a pre-processing step, it can easily be integrated into many machine translation systems. On the other hand, reordering decisions are made independent of the other features in the decoder, such as the language model. If an incorrect permutation is selected, it is not easy for the decoder to undo it.

Another category of reordering models, called lexicalised reordering, can be integrated into the decoder as an additional feature or features, so the reordering scores are combined with evidence provided by other features. Lexicalised reordering models were first introduced by [6]. They condition the reordering on the previously translated phrase and the next phrase to be translated considering the source and target sides. Different movements are grouped together to deal with data sparsity. [7] conditioned the exact jumps on the source side words (unigram) and had three features added to the decoder. [8] considered both source side and target side phrases and predicted three different types of movements of the phrases[1]. [9] argue that previous lexicalised reordering models fail capturing long distance reorderings and propose a hierarchical lexicalised reordering model. Despite dealing with hierarchical reordering rules, their method does not rely on cubic-time parsing algorithms such as those used in hierarchical phrase-based models ([10]). The model analyses the alignments beyond adjacent phrases to extract reordering rules, which are more complex than predicting the orientation between blocks of consecutive phrases. They classify lexicalised reordering models into word-based, phrase-based and hierarchical orientation models and demonstrate that the latter performs significantly better than the others.

[11] have considered reordering in machine translation as a case of Linear Ordering Problem and learned the relative orders of words in a sentence based on multiple features. A dynamic programming algorithm based on chart parsing is developed to find the best reordering within a neighbourhood. They have used the method as a preprocessing step to translate German to English and reported improvements over a strong baseline equipped with a lexicalised reordering model.

[12] proposed a method based on maximum entropy principle to combine different features and predict the word orientation. They combined multiple lexicalised features and for generalisation, considered features based on word classes. They concluded that features based on the source sentence words perform better than those based on the target side and allowing for more context always helps. Since predicting the exact position is not easy, the next positions are grouped together and the model predicts the class of the next jump. Although, they only report the results for a small set of classes (backward, monotone and forward), their model is general enough to predict more fine-grained classes. Inspired by their work, [13] have built two models for each transition. One based on the features of the outbound word (the word that has just been translated) and one model based on features of the inbound word (the word, we are about to translate). Their feature set includes words, part of speech (POS) tags and sentence length features. They argue that using the new models renders the linear future distortion cost inappropriate and add future distortion cost as another feature to be optimised through MERT. In [14] a maximum entropy based model is proposed to predict the orientation of neighbouring blocks in their BTG[2]-based decoder. They have two types of BTG merging rules, straight or inverted and the reordering model weights the merging rules using lexicalised features of the source and target side. Following [15], they extend the model to include linguistically-aware features.

With the same motivation as ours, that different sentence types require different reordering treatments, [16] classify the Chinese sentences under three categories and build reordering models for each category. For sentence type identification, a Support Vector Machine (SVM) classifier is built, with features including all the words in the sentence. They report substantial improvements over the baseline for the Chinese-to-English IWSLT 2007 task.

[1]The model is implemented in the open source SMT system, Moses `http://www.statmt.org/moses`. It is possible to configure the system to build the model with different contexts. For example, only source side or only previous phrases.

[2]Bracketing Transduction Grammar

3. Distortion and Translation Quality

As mentioned before, due to the complex nature of decoding in machine translation [17], many parameters are used to manage the size of the search space. Distortion limit or the skip widow size is one of the most important parameters that controls the freedom of the decoder in permuting words to capture the word order differences between the source and the target languages. The best results on different language pairs need different settings for the distortion limit. It is common to set the parameter according to the nature of languages involved and with respect to speed and memory requirements. Longer limits lead the decoder to generate more hypotheses and increase translation time. However, increase in time is not the only drawback of having a longer distortion limit. More hypotheses are generated, therefore more burden is put on the language model to choose the best reordering decision.

Figure 2 shows the result of decoding with distortion limits between 1 and 15. Although both graphs show the results of an identical system on two data-sets, the best result for each one of them is achieved by different parameters. One way to find the best distortion limit is to run the tuning process with a range of distortion limits and choose the one with the highest score. Apart from the substantial amount of work required to perform the tuning several times, it is not even guaranteed that the best distortion limit for the development set is the best for the unseen test set.

Another parameter related to distortion is the reordering constraint strategy, which controls the decoder in how to skip words and return back for open positions. [18] investigated different reordering constraints and reported their differences on multiple translation tasks. [19] also proposed a method to find the best reordering constraint independent of other features and solely based on the ability of the constraint to cover all the needed n-grams in a sentence. Figure 3 shows the translation quality for two different reordering constraints on a Turkish-to-English translation task. One graph of figure 3 is constrained by the so-called "Window length" constraint, which restricts the decoder by not letting it to choose a phrase with more than dl words distance from the first open position of the source sentence. The constraint in the other graph is "Maximum distortion", which is more relaxed and the only restriction is the distance between the last translated phrase and the next one [20]. As one can see, the Turkish-English language pair requires relatively long distortion limits, however, the maximum distortion strategy reaches the best results earlier than the window length strategy and overall has a higher score.

We propose a method of selecting the best distortion limit in each step of hypothesis expansion. This method determines the size of the window required to be searched for the next phrase to be translated. Adjusting the distortion limit prevents the decoder to explore undesirable parts of the search space. This saves both time and improves the performance by avoiding extra noise during the search. In the next

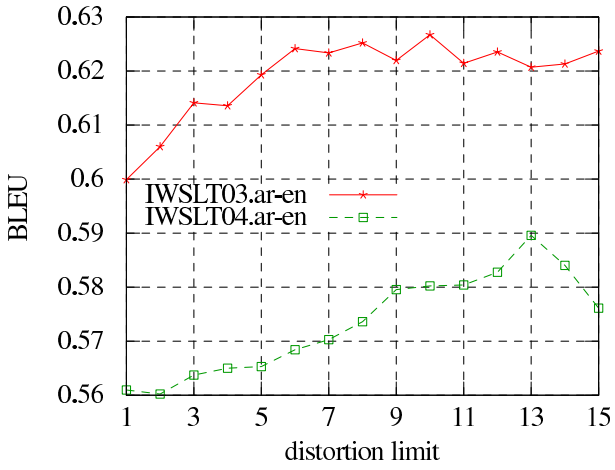

Figure 2: The effect of the distortion limit parameter on the quality of the translation system. Both graphs are results of the baseline system (see Section 6) on Arabic-English of BTEC task, tuned on IWSLT03.ar-en.

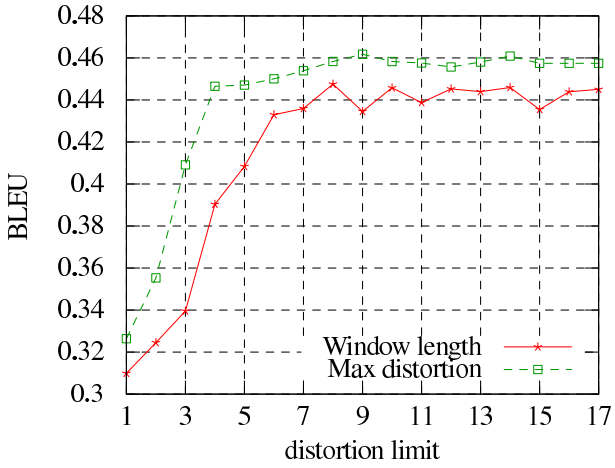

Figure 3: Results of two different reordering constraints on the Turkish-English of the BTEC task. Both graphs are the BLEU score of the baseline system on the IWSLT03.tr-en tuning set.

section, we first describe a lexicalised reordering model to establish the main set of features required for a discriminative reordering model.

4. Reordering Models

4.1. Lexicalised Reordering Model

We build a lexicalised reordering model based on [7] with three additional features modelling the costs of jumping from, jumping to and jumping over the words involved in

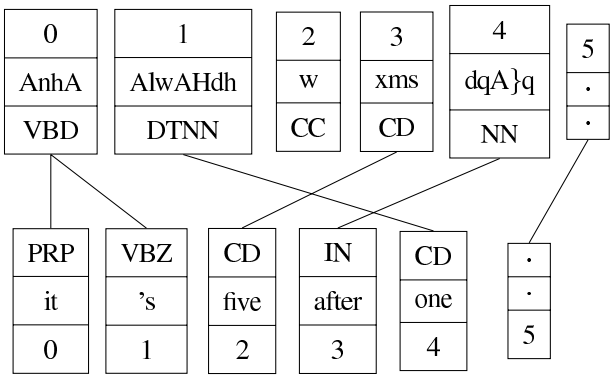

Figure 4: A word alignment example of a sentence from the Arabic-English training data. The Arabic sentence is romanised according to Buckwalter's method.

the reordering. Assume we want to collect training frequencies from the example sentence in figure 4. We loop over the target sentence and collect the jump statistics by considering e_i and e_{i+1}, where $0 <= i < I$. For example, for $i = 1$, we consider e_1 and e_2, which are aligned to f_0 and f_3 respectively. The following words are the local context of this jump (from f_0 to f_3) and their respected frequencies will be increased by one:

1. f_0 as outbound word
2. f_3 as inbound word
3. f_1 and f_2 as jumped over words

To avoid collecting evidence for a jumped over word multiple times, the frequency of being jumped over for a position only increases once. We collect the above frequencies for all the jumps in one sentence and all the sentences in the training data.

The training examples defined above will be used to add three additional features to the decoder:

$$l_o(f_1^J, j, j', d_{j,j'}) = \frac{\text{count}_o(f_j, d_{j,j'})}{\text{count}(f_j)} \quad (1)$$

which is smoothed by a factor (α) as:

$$l_o(f_1^J, j, j', d_{j,j'}) = \frac{\alpha \frac{\text{count}_o(d_{j,j'})}{\sum_{d \in D} \text{count}_o(d)} + \text{count}_o(f_j, d_{j,j'})}{\text{count}_o(f_j)} \quad (2)$$

where, $d_{j,j'}$ is a class associated with a range that the distance between j and j' belongs to. D is the set of all jump classes. The distance between j and j' is defined as:

$$\text{distance}(j, j') = \begin{cases} j - j' - 1 & \text{if } j \geq j' \\ j' - j & \text{if } j < j' \end{cases} \quad (3)$$

Two more features l_i (inbound) and l_j (jumped-over), similar to this are also added for inbound and jumped-over words.

We performed a small series of experiments to evaluate the effect of these features on the translation quality and the system equipped with these features improved the baseline (see Section 6) for the two best performing distortion limits of the baseline. Table 1 shows the results.

SET	RUN	DL=6	DL=10
IWSLT08(dev)	BASELINE	0.5348	0.5449
	LEX	**0.5461**	**0.5534**
IWSLT07(test)	BASELINE	0.5022	0.5128
	LEX	**0.5121**	**0.5142**

Table 1: Comparing the baseline and the lexicalised reordering model with inbound, outbound and jumped-over features. The results are on Arabic-English of BTEC task.

4.2. Discriminative Reordering Model

The results in the previous section show that the distance-based distortion penalty plus the language model are not enough for making the best reordering decisions. Lexicalised reordering models [6, 21] have been shown to be effective for many language pairs in improving the translation quality. However, because we want to predict the distortion limit, we need to calculate all the reordering costs before decoding the sentence. Additionally, we want to incorporate features extracted from the whole sentence, along with surface features of the phrases we are about to translate in the reordering model. Lexicalised reordering models rely on surface forms of the source and target phrases that have been translated or the ones we are about to translate. Factored models [22] have been proposed to incorporate features such as POS-tags, however, global features such as chunk information are not easily included.

Inspired by [12], we build a maximum entropy classifier [23] that predicts the length of the next jump based on the local lexicalised features and the sentence structure. To increase the classification accuracy, we divide the jumps into a set of classes. For example, jumps with length 2 to 4 are in one class, those with length 5 to 9 in another, etc. Feature functions are binary functions of the form:

$$h_k(f_1^J, j, j', d_{j,j'}) \quad (4)$$

where, f_1^J is the source sentence with all the syntactic information including POS and chunking tags. h_k is a binary function which is 1 when the feature is present for the specific jump decision and 0, if it is not. j and j' are source positions and $d_{j,j'}$ is the jump class between them. The decision formula is:

$$p(d_{j,j'}|f_1^J, j, j') = \frac{1}{Z} \exp\left(\sum_{k=1}^{N} \lambda_k h_k(f_1^J, j, j', d_{j,j'})\right) \quad (5)$$

where Z is a normalisation factor:

$$Z = \sum_{d \in D} \exp\left(\sum_{k=1}^{N} \lambda_k h_k(f_1^J, j, j', d)\right) \quad (6)$$

One of the main benefits of using a discriminative model for this classification task is the ability of these models to learn millions of inter-dependent features. We define an extensive set of features including mostly local context of each jump and some of the characteristics of the sentence. The following list is the set of features used in training the model for a jump from j to j' in sentence f_1^J:

- inbound (IN) and outbound (OUT) words, f_j and $f_{j'}$
- both words together (PAIR), $f_j + f_{j'}$
- jumped over (OVER) words, all the words between j and j' as described in Section 4.1
- part of speech tags of inbound, outbound, pairwise and jumped over words (IN.POS, OUT.POS and ...)
- bigram inbound (IN2) and outbound (OUT2), $f_{j-1}+f_j$ and $f_{j'} + f_{j'+1}$
- are both j and j' in the same syntactic chunk or not (1CHUNK and 2CHUNK)?
- does f_1^J contain a question mark (IS.Q)?
- is there a question mark or full stop between j and j' (CROSS.FULL)?
- is there a punctuation mark between j and j' (CROSS.PUNCT)?

Table 2 shows the contribution of each set of features to the quality of the model. We used the Arabic-English training data for these experiments. 500 sentences were heldout for validation and 500 sentences were set aside for testing. The rest of the collection was used for training the model.

Features	Accuracy	F_1^M	$\hat{\pi}^M$	$\hat{\rho}^M$
OUT, IN	0.7127	0.5306	0.6538	0.4935
+OVER	0.8337	0.6265	0.7720	0.5874
+PAIR	0.8460	0.6617	0.7940	0.6197
+(*.POS)	0.8826	0.6909	0.8496	0.6503
+(*.POS2)	0.9024	0.7666	0.8392	0.7290
+IS.Q,CROSS.*	0.9042	0.7806	0.8525	0.7404
+IN2,OUT2	0.9085	**0.7964**	0.8643	**0.7566**
ALL	**0.9091**	0.7958	**0.8737**	0.7503

Table 2: Classification results of the maximum entropy classifier with different features and the contribution of each set of features. F_1^M, $\hat{\pi}^M$ and $\hat{\rho}^M$ are macro F-measure, macro-precision and macro-recall respectively. macro F-measure is calculated by averaging over the F-measures of each class. *.POS means all the features that their name end with .POS. The evaluation is done on the Arabic-English data set.

5. Dynamic Distortion

In Section 3, we argued for the importance of determining the optimum distortion limit. Both translation quality and decoding speed are influenced by changing this parameter. The discriminative model described in the previous section, provides us with some information about the reordering needs of a sentence before starting to decode it. This enables us to determine the best distortion limit for this particular sentence and this particular hypothesis expansion.

Changing the distortion limit for each sentence or more specifically for each hypothesis expansion, has a few advantages: Firstly, it removes the need for tuning the system with many different distortion limit settings to find the best one. As it is clear from the results of Section 3, the best value for the parameter on one data set may not be the best for another. Secondly, the limit can be very long for some sentences or some parts of a sentence. Changing it for each hypothesis expansion can compensate for long distortion in terms of decoding speed. Basically, we increase the distortion when it is needed and save time when there is no need for long distance reorderings. Thirdly, adjusting the distortion limit reduces the amount of unnecessary jumps in some parts of the sentence and hence decreases noise in the search process, which leads to better translation quality. Additionally, other parameters of the search algorithm that control the size of the search space, such as beam width or stack size can be increased without increasing the decoding time substantially.

Before decoding sentence f_1^J, we use the classifier described in the previous section to compute the probability $p(d_{j,j'}|f_1^J, j, j')$ for each j and j', where $0 <= j, j' <= J+1$ and for all $d \in D$. 0 and $J+1$ are also considered to include the initial move after the start and the final jump before the end symbol. In the next step, the most probable jump after each source position is calculated and the distance is saved as the best distortion limit after that position. To score the jumps after each source position j, equation 7 is used:

$$s_j(j') = \prod_{j''=0}^{j''=j'} p(d_{j,j''}|f_1^J, j, j'') \prod_{j''=j'+1}^{j''=J+1} (1-p(d_{j,j''}|f_1^J, j, j'')) \quad (7)$$

and the distortion limit estimated by this approach for position j equals to:

$$dl(j) = \text{distance}(j, \arg\max_{j'}\{s_j(j')\}) \quad (8)$$

where distance is defined in equation 3. This way we find the most likely jump after f_j and set the distortion limit at position j to length of the jump. The above equations are for forward distortion and similar equations are used for backward distortions.

Data set	Source lang	Sentences	Average. len	Words	Vocabulary	OOV	Number of refs
train	Arabic	19972	8.50	169943	14519	-	-
train	Turkish	19972	8.12	162198	6098	-	-
IWSLT03.ar-en	Arabic	506	6.56	3323	1095	111(3.34%)	16
IWSLT04.ar-en	Arabic	500	6.95	3479	1189	101(2.90%)	16
IWSLT05.ar-en	Arabic	506	6.66	3375	1182	124(3.67%)	16
IWSLT07.ar-en	Arabic	489	6.45	3158	1100	165(5.22%)	6
IWSLT08.ar-en	Arabic	507	6.73	3414	1130	153(4.48%)	16
IWSLT03.tr-en	Turkish	506	6.18	3131	1142	152(4.85%)	16
IWSLT04.tr-en	Turkish	500	6.19	3096	1209	175(5.65%)	16

Table 3: Corpus statistics and OOV token rates for the development and test sets used for the experiments.

6. Experiments

To examine the effects of the discriminative reordering model and the dynamic distortion on translation quality, we have chosen the Arabic-to-English and Turkish-to-English data sets from the IWSLT BTEC task as they involve many short, medium, and long distance re-orderings. Some of the statistics of the data sets are shown in Table 3.

6.1. Baseline

The preprocessing stage for Arabic-to-English includes tokenisation of both sides and lower casing of the English side. We removed all the diacritic characters from the Arabic side and normalised punctuation. For tokenising Turkish, we used Morfessor [24] to automatically analyse the morphology of the source side. Lower casing was applied to both source and target sides of Turkish and English.

The decoder is a common multi-beam, multi-stack phrase-based decoder, described in [25] with the following features:

- phrase translation probabilities and lexical probabilities for both directions
- a 4-gram language model
- phrase and word penalties
- distance-based re-ordering penalty

The weights for the features are optimised by MERT [26] to maximise the BLEU [27] score. We optimised the discriminative model using the L-BFGS implementation within the `MALLET` toolkit [28]. The built model is used to score the reordering options before the decoding.

6.2. Results

For the Turkish-to-English task, we tune the baseline (BASELINE) and the discriminative reordering model (DISCRIM-REO) for distortion limits 0 to 17 and tune Arabic-English for distortion limits 0 to 15. For both tasks dynamic distortion method (DYNAMIC-DL) is tuned. Tables 4 and 5 show the results for the Turkish-to-English and Arabic-to-English tasks, respectively. For both tasks we ran the baseline with the lexicalised reordering model of `Moses` [8], with no significant improvements, so we did not include the results of the lexicalised reordering model here.

In the Arabic-to-English task the window length constraint performs better than the other constraints. In this constraint the size of the jump is restricted by the first uncovered position of the source sentence. However, since we change the distortion limit during decoding for the dynamic distortion method, an uncovered position outside the window for one move can be inside the distortion limit window for another. Therefore, we relax this restriction in the dynamic distortion method and allow the decoder to make jumps, even if the first uncovered position remains outside the current distortion. Also, we relax the backward distortion limit restriction if there is an uncovered position outside it.

In most cases, DISCRIM-REO performs better than the baseline, particularly on longer distortion limits, which is expected given the fact it has an extra feature to deal with the large amount of reordering decisions. In all the experiments, confirming previous findings [13], we found that the future distortion cost is crucial for the quality of the translation, particularly for systems with long distortion parameters.

Overall the discriminative model and the dynamic distortion method performed better for Turkish-to-English compared to Arabic-to-English. This can be justified by the fact that Turkish-to-English translation requires more reorderings than Arabic to English.

7. Conclusions

We showed that choosing the best distortion limit for a language pair or even a data set can gain substantial improvements in phrase-based statistical machine translation decoders. To avoid the difficulty of running with all possible settings, we proposed a method of dynamically adjusting the distortion limit for each hypothesis expansion in phrase-based decoders. To determine the best value for the distortion limit at each move, a discriminative reordering model with numerous features is built and integrated into the decoder as an extra feature.

Results of the experiments by DISCRIM-REO show that

SET	RUN	DL=6	DL=11	DL=17
IWSLT03(dev)	BASELINE	0.4500	0.4576	0.4574
	DISCRIM-REO	0.4591	**0.4641**	**0.4669**
	DYNAMIC-DL	**0.4640**	0.4640	0.4640
IWSLT04(test)	BASELINE	0.4273	0.4366	0.4363
	DISCRIM-REO	0.4378	0.4434	0.4412
	DYNAMIC-DL	**0.4492**	**0.4492**	**0.4492**

Table 4: Experimental results on Turkish-English data sets. The first three rows show the result on the development set and the rest of the results on the test set. set.

SET	RUN	DL=3	DL=6	DL=9	DL=12	DL=15
IWSLT08(dev)	BASELINE	0.5358	0.5348	0.5464	0.5383	0.5416
	DISCRIM-REO	0.5338	0.5458	0.5507	0.5489	0.5489
	DYNAMIC-DL	**0.5571**	**0.5571**	**0.5571**	**0.5571**	**0.5571**
IWSLT03(test)	BASELINE	0.6001	0.6024	0.6199	0.6076	0.6129
	DISCRIM-REO	0.6034	0.6053	0.6220	0.6123	0.6137
	DYNAMIC-DL	**0.6228**	**0.6228**	**0.6228**	**0.6228**	**0.6228**
IWSLT04(test)	BASELINE	0.5619	0.5733	0.5765	0.5789	0.5784
	DISCRIM-REO	0.5534	0.5748	0.5794	0.5820	0.5844
	DYNAMIC-DL	**0.5856**	**0.5856**	**0.5856**	**0.5856**	**0.5856**
IWSLT05(test)	BASELINE	0.5789	0.5875	0.5966	0.5841	0.6007
	DISCRIM-REO	0.5815	0.5922	0.6002	0.5941	0.5853
	DYNAMIC-DL	**0.6016**	**0.6016**	**0.6016**	**0.6016**	**0.6016**
IWSLT07(test)	BASELINE	0.5010	0.5022	0.5103	0.5130	0.5098
	DISCRIM-REO	0.5047	0.5091	0.5196	0.5136	0.5141
	DYNAMIC-DL	**0.5242**	**0.5242**	**0.5242**	**0.5242**	**0.5242**

Table 5: Experiment results on Arabic-English data sets. The first three rows show the result on the development set and the rest of the results on the test set.

more features in the discriminative reordering model helps to improve the accuracy of the classification and the quality of the translation, however, lexical features are more effective than POS or chunk-based features.

Since there is no difference between the features of DISCRIM-REO and DYNAMIC-DL, the improvements achieved by the latter is due to the change of the search space explored by the decoder. Therefore, guiding the decoder during the search can be effective in improving the quality of translation.

8. Acknowledgements

This work has been funded in part by the European Commission through the CoSyne project FP7-ICT-4-248531 and the GALATEAS project CIP-ICT PSP-2009-3-250430.

9. References

[1] S. DeNeefe, K. Knight, W. Wang, and D. Marcu, "What can syntax-based MT learn from phrase-based MT?" in *Proceedings of the 2007 Joint Conference on Empirical Methods Innatural Language Processing and Computational Natural Language Learning(emnlp-Conll)*, 2007, pp. 755–763.

[2] F. Xia and M. McCord, "Improving a statistical MT system with automatically learned rewrite patterns," in *COLING '04: Proceedings of the 20th International Conference on Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2004, p. 508.

[3] M. Collins, P. Koehn, and I. Kučerová, "Clause restructuring for statistical machine translation," in *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2005, pp. 531–540.

[4] C. Wang, M. Collins, and P. Koehn, "Chinese syntactic reordering for statistical machine translation," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (emnlp-Conll)*, 2007, pp. 737–745.

[5] I. Badr, R. Zbib, and J. R. Glass, "Syntactic phrase reordering for English-to-Arabic statistical machine translation," in *EACL*, 2009, pp. 86–93.

[6] C. Tillmann, "A unigram orientation model for statistical machine translation," in *HLT-NAACL 2004: Short Papers*, D. M. Susan Dumais and S. Roukos, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 101–104.

[7] Y. Al-Onaizan and K. Papineni, "Distortion models for sta-

tistical machine translation," in *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL*. Morristown, NJ, USA: Association for Computational Linguistics, 2006, pp. 529–536.

[8] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, D. Talbot, and M. White, "Edinburgh system description for the 2005 NIST MT evaluation," in *MT Eval Workshop 2005*, 2005.

[9] M. Galley and C. D. Manning, "A simple and effective hierarchical phrase reordering model," in *Proceedings of EMNLP*, 2008.

[10] D. Chiang, "A hierarchical phrase-based model for statistical machine translation," in *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2005, pp. 263–270.

[11] R. Tromble and J. Eisner, "Learning linear ordering problems for better translation," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, August 2009, pp. 1007–1016.

[12] R. Zens and H. Ney, "Discriminative reordering models for statistical machine translation," in *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL): Proceedings of the Workshop on Statistical Machine Translation*, New York City, NY, June 2006, pp. 55–63.

[13] S. Green, M. Galley, and C. D. Manning, "Improved models of distortion cost for statistical machine translation," in *NAACL*, 2010.

[14] D. Xiong, Q. Liu, and S. Lin, "Maximum entropy based phrase reordering model for statistical machine translation," in *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2006, pp. 521–528.

[15] D. Xiong, M. Zhang, A. Aw, and H. Li, "A linguistically annotated reordering model for BTG-based statistical machine translation," in *Proceedings of ACL-08: HLT, Short Papers*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 149–152.

[16] J. Zhang, C. Zong, and S. Li, "Sentence type based reordering model for statistical machine translation," in *COLING '08: Proceedings of the 22nd International Conference on Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2008, pp. 1089–1096.

[17] K. Knight, "Decoding complexity in word-replacement translation models," *Comput. Linguist.*, vol. 25, no. 4, pp. 607–615, 1999.

[18] R. Zens, H. Ney, T. Watanabe, and E. Sumita, "Reordering constraints for phrase-based statistical machine translation," in *COLING '04: Proceedings of the 20th International Conference on Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2004, p. 205.

[19] M. Dreyer, K. Hall, and S. Khudanpur, "Comparing reordering constraints for SMT using efficient BLEU oracle computation," in *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*. Rochester, New York: Association for Computational Linguistics, April 2007, pp. 103–110.

[20] A. Lopez, "Translation as weighted deduction," in *EACL*, 2009, pp. 532–540.

[21] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh system description for the 2005 IWSLT speech translation evaluation," in *International Workshop on Spoken Language Translation*, 2005.

[22] H. Hoang and P. Koehn, "Improving mid-range re-ordering using templates of factors," in *EACL*, 2009, pp. 372–379.

[23] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Comput. Linguist.*, vol. 22, no. 1, pp. 39–71, 1996.

[24] M. Creutz and K. Lagus, *Unsupervised Morpheme Segmentation and Morphology Induction from Text Corpora Using Morfessor 1.0*, March 2005, ch. Report A81.

[25] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 48–54.

[26] F. J. Och, "Minimum error rate training in statistical machine translation," in *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 160–167.

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2001, pp. 311–318.

[28] A. K. McCallum, "MALLET: A machine learning for language toolkit," 2002, http://mallet.cs.umass.edu.