Accuracy-Based Scoring for DOT: Towards Direct Error Minimization for Data-Oriented Translation

Daniel Galron
CIMS
New York University
galron@cs.nyu.edu

Sergio Penkale, Andy Way
CNGL
Dublin City University
{spenkale,away}
@computing.dcu.ie

I. Dan Melamed
AT&T Shannon Laboratory
{lastname}
@research.att.com

Abstract

In this work we present a novel technique to rescore fragments in the Data-Oriented Translation model based on their contribution to translation accuracy. We describe three new rescoring methods, and present the initial results of a pilot experiment on a small subset of the Europarl corpus. This work is a proof-of-concept, and is the first step in directly optimizing translation decisions solely on the hypothesized accuracy of potential translations resulting from those decisions.

1 Introduction

The Data-Oriented Translation (DOT) (Poutsma, 2000) model is a tree-structured translation model, in which linked subtree fragments extracted from a parsed bitext are composed to cover a source-language sentence to be translated. Each linked fragment pair consists of a source-language side and a target-language side, similar to (Wu, 1997). Translating a new sentence involves composing the linked fragments into derivations so that a new source-language sentence is covered by the source tree fragments of the linked pairs, where the yields of the target-side derivations are the candidate translations. Derivations are scored according to their likelihood, and the translation is selected from the derivation pair with the highest score. However, we have no reason to believe that maximizing likelihood is the best way to maximize translation accuracy – likelihood and accuracy do not necessarily correlate well.

We can frame the problem as a search problem, where we are searching a space of derivations for the one that yields the highest scoring translation. By putting weights on the derivations in the search space, we wish to point the decoder in the direction of the optimal translation. Since we want the decoder to find the translation with the highest evaluation score, we would want to score the derivations with weights that correlate well with the particular evaluation measure in mind.

Much of the work in the MT literature has focused on the scoring of translation decisions made. (Yamada and Knight, 2001) follow (Brown et al., 1993) in using the noisy channel model, by decomposing the translation decisions modeled by the translation model into different types, and inducing probability distributions via maximum likelihood estimation over each decision type. This model is then decoded as described in (Yamada and Knight, 2002). This type of approach is also followed in (Galley et al., 2006).

There has been some previous work on accuracy-driven training techniques for SMT, such as MERT (Och, 2003) and the Simplex Armijo Downhill method (Zhao and Chen, 2009), which tune the parameters in a linear combination of various phrase scores according to a held-out tuning set. While this does tune the relative weights of the scores to maximize the accuracy of candidates in the tuning set, the scores themselves in the linear combination are not necessarily correlated with the accuracy of the translation. Tillmann and Zhang (2006) present a procedure to directly optimize the global scoring function used by a phrase-based decoder on the accuracy of the translations. Similarly to MERT, Tillmann and Zhang estimate the parameters of a weight vector on a linear combination of (binary) features using a global objective function correlated with BLEU (Papineni et al., 2002).

In this work, we prototype some methods for moving directly towards incorporating a measure of the translation quality of each fragment used, bringing DOT more into the mainstream of current SMT research. In Section 2 we describe probability-based DOT fragment scoring. In Section 3 we describe our rescoring setup and the

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 371–380,
Singapore, 6-7 August 2009.

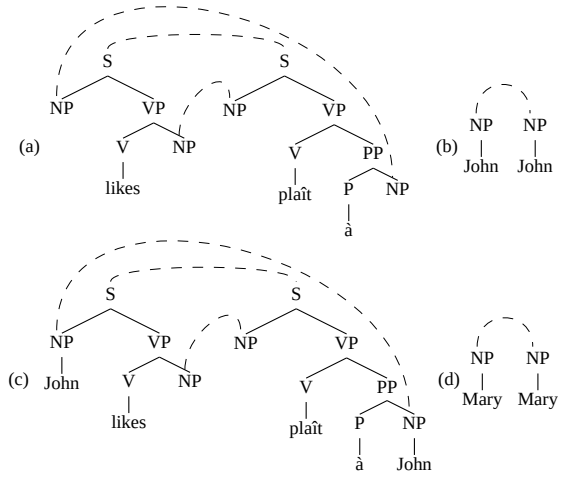


Figure 1: Example DOT Fragments.

three rescoring methods. In Section 4, we describe our experiments. In Section 5 we compare the results of rescoring the fragments with the three methods. In Section 6 we discuss some of the decisions that are affected by our rescoring methods. Finally, we discuss the next steps in training the DOT system by optimizing over a translation accuracy-based objective function in Section 7.

2 DOT Scoring

As described in previous work (Poutsma, 2000; Hearne and Way, 2003), DOT scores translations according to the probabilities of the derivations, which are in turn computed from the relative frequencies of linked tree fragments in a parallel treebank. Linked fragment pairs are conditionally independent, so the score of a derivation is the product of the probabilities of all the linked fragments used. To find the probability of a translation, DOT marginalizes over the scores of all derivations yielding the translation.

From a parallel treebank aligned at the subsentential level, we extract all possible linked fragment pairs by first selecting all linked pairs of nodes in the treebank to be the roots of a new subtree pair, and then selecting a (possibly empty) set of linked node pairs that are descendants of the newly selected fragment roots and deleting all subtree pairs dominated by these nodes. Leaves of fragments can either be terminals, or non-terminal *frontier nodes* where we can compose other fragments (c.f. (Eisner, 2003)). We give example DOT fragment pairs in Figure 1.

Given two subtree pairs $\langle s_1, t_1 \rangle$ and $\langle s_2, t_2 \rangle$, we can compose them using the DOT composition operator $\circ$ if the leftmost non-terminal frontier node of s_1 is equal to the root node of s_2, and the leftmost non-terminal frontier node of s_1's *linked counterpart* in t_1 is equal to the root node of t_2. The resulting tree pair consists of a copy of s_1 where s_2 has been inserted at the leftmost frontier node, and a copy of t_1 where t_2 has been inserted at the node linked to s_1's leftmost frontier node (Hearne and Way, 2003).

In Figure 1, fragment pair (a) is a fragment with two open substitution sites. If we compose this fragment pair with fragment pair (b), the source side composition must take place on the leftmost non-terminal frontier node (the leftmost NP). On the target side we compose on the frontier linked to the leftmost source side non-terminal frontier. The result is fragment pair (c). If we now compose the resulting fragment pair with fragment pair (d), we obtain a fragment pair with no open substitution sites whose source-side yield is *John likes Mary* and whose target-side yield is *Mary plaît à John*. Note that there are two different derivations using the fragment pairs in Figure 1 that result in the same fragment pair, namely (a) $\circ$ (b) $\circ$ (d), and (c) $\circ$ (d).

For a given linked fragment pair $\langle d_s, d_t \rangle$, the probability assigned to it is

$$P(\langle d_s, d_t \rangle) = \frac{|\langle d_s, d_t \rangle|}{\sum_{r(u_s)=r(d_s) \wedge r(u_t)=r(d_t)} |\langle u_s, u_t \rangle|} \quad (1)$$

where $|\langle d_s, d_t \rangle|$ is the number of times the fragment pair $\langle d_s, d_t \rangle$ is found in the bitext, and $r(d)$ is the root nonterminal of d. Essentially, the probability assigned to the fragment pair is the relative frequency of the fragment pair to the pair of nonterminals that root the fragments.

Then, with the assumption that DOT fragments are conditionally independent, the probability of a derivation is

$$\begin{aligned} P(\mathbf{d}) &= P(\langle d_s, d_t \rangle_1 \circ \ldots \circ \langle d_s, d_t \rangle_N) \\ &= \prod_i P(\langle d_s, d_t \rangle_i) \end{aligned} \quad (2)$$

In the original DOT formulation, DOT disambiguated translations according to their probabilities. Since a translation can have many possible derivations, to obtain the probability of a translation it is necessary to marginalize over the distinct derivations yielding a translation. The probability of a translation w_t of a source sentence w_s, is

given by (3):

$$P(w_s, w_t) = \sum_{\mathbf{d} \in \mathbf{D}} P(\mathbf{d}_{\langle w_s, w_t \rangle}) \quad (3)$$

and the translation is chosen so as to maximize (4):

$$\hat{w_t} = \underset{w_t}{\operatorname{argmax}} P(w_s, w_t) \quad (4)$$

Hearne and Way (2006) examined alternative disambiguation strategies. They found that rather than disambiguating on the translation probability, the translation quality would improve by disambiguating on the derivation probability, as in (5):

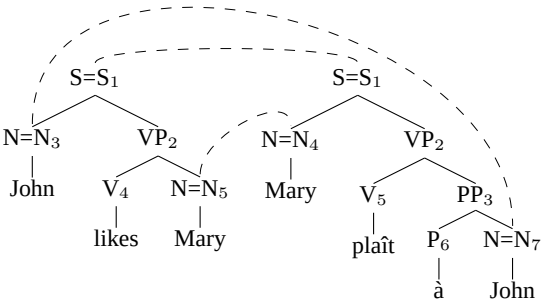
$$\hat{w_t} = \underset{\mathbf{d}}{\operatorname{argmax}} P(\mathbf{d}) \quad (5)$$

Our analysis suggest that this is because many derivations with very low probabilities generate the same, poor translation. When applying Equation (3) to marginalize over those derivations, the resulting score is higher for the poor translation than a better translation with fewer derivations but where the derivations had higher likelihood.

Using the DOT model directly is difficult – the number of fragments extracted from a parallel treebank is exponential in the size of the treebank. Therefore we use the Goodman reduction of DOT (Hearne, 2005) to create an isomorphic PCFG representation of the DOT model that is linear in the size of the treebank. The idea behind the Goodman reduction is that rather than storing fragments in the grammar and translating via composition, we simultaneously build up the fragments using the PCFG reduction and compose them together. To perform the reduction, we first relabel the two linked nodes (X, Y) with the new label X=Y. We then label each node in the parallel treebank with a unique Goodman index. Each binary-branching node and its two children can be internal or root/frontier. We add rules to the grammar reflecting the role that each node can take, keeping unaligned nodes as fragment-internal nodes. So in the case where a node and both of its children are aligned, we commit 8 rules into the grammar, as follows:

LHS → RHS1 RHS2	LHS+a → RHS1 RHS2
LHS → RHS1+b RHS2	LHS+a → RHS1+b RHS2
LHS → RHS1 RHS2+c	LHS+a → RHS1 RHS2+c
LHS → RHS1+b RHS+c	LHS+a → RHS1+b RHS2+c

A category label which ends in a '+' symbol followed by a Goodman index is fragment-internal and all other nodes are either fragment roots or



Source PCFG		Target PCFG	
S=S → N=N VP+2	0.5	S=S → N=N VP+2	0.5
S=S → N=N+3 VP+2	0.5	S=S → N=N+4 VP+2	0.5
S=S+1 → N=N VP+2	0.5	S=S+1 → N=N VP+2	0.5
S=S+1 → N=N+3 VP+2	0.5	S=S+1 → N=N+4 VP+2	0.5
N=N → John	0.5	N=N → Mary	0.5
N=N+3 → John	1	N=N+4 → Mary	1
VP+2 → V+4 N=N	0.5	VP+2 → V+5 PP+3	1
VP+2 → V+4 N=N+5	0.5	V+5 → plaît	1
V+4 → likes	1	PP+3 → P+6 N=N	0.5
N=N → Mary	0.5	PP+3 → P+6 N=N+7	0.5
N=N+5 → Mary	1	P+6 → à	1
		N=N → John	0.5
		N=N+7 → John	1

Figure 2: A parallel tree and its corresponding Goodman reduction.

frontier nodes. A fragment pair, then, is a pair of subtrees in which the root does not have an index, all internal nodes have indices, and all the leaves are either terminals or un-indexed nodes. We give an example Goodman reduction in Figure 2.

While we store the source grammar and the target grammar separately, we also keep track of the correspondence between source and target Goodman indices and can easily identify the alignments according to the Goodman indices. Probabilities for the PCFG rules are computed monolingually as in the standard Goodman reduction for DOP (Goodman, 1996). In decoding with the Goodman reduction, we first find the *n*-best parses on the source side, and for each source fragment, we construct the *k*-best fragments on the target side. We finally compute the bilingual derivation probabilities by multiplying the source and target derivation probabilities by the target fragment relative frequencies conditioned on the source fragment.

There are a few problems with a likelihood-based scoring scheme. First, it is not clear that if a fragment is more likely to be seen in training data then it is more likely to be used in a correct translation of an unseen sentence. In our analysis of the candidate translations of the DOT system, we observed that frequently, the highest-likelihood candidate translation output by the system was not the highest-accuracy candidate inferred. An additional problem is that, as described in (Johnson, 2002), the relative frequency estimator for DOP

(and by extension, DOT) is known to be biased and inconsistent.

3 Accuracy-Based Fragment Scoring

In our work, we wish to incorporate a measure of fragment accuracy into the scoring. To do so, we reformulate the scoring of DOT as log-linear rather than probabilistic, in order to incorporate non-likelihood features into the derivation scores. For all tree fragment pairs $\langle d_s, d_t \rangle$, let

$$l(\langle d_s, d_t \rangle) = \log(p(\langle d_s, d_t \rangle)) \quad (6)$$

The general form of a rescored tree fragment will be

$$s(\langle d_s, d_t \rangle) = \alpha_0 l(\langle d_s, d_t \rangle) + \sum_{i=1}^{k} \alpha_i f_i(\langle d_s, d_t \rangle) \quad (7)$$

where each α_i is the weight of that term in the final score, and each $f_i(d)$ is a feature. In this work, we only consider $f_1(d)$, an accuracy-based score, although in future work we will consider a wide variety of features in the scoring function, including combinations of the different scoring schemes described below, binary lexical features, binary source-side syntactic features, and local target side features. The score of a derivation is now given by (8):

$$\begin{aligned} s(d) &= s(\langle d_s, d_t \rangle_1 \circ \ldots \circ \langle d_s, d_t \rangle_N) \\ &= \sum_i s(\langle d_s, d_t \rangle_i) \end{aligned} \quad (8)$$

In order to disambiguate between candidate translations, we follow (Hearne and Way, 2006) by using Equation (5).

3.1 Structured Fragment Rescoring

In all our approaches, we rescore fragments according to their contribution to the accuracy of a translation. We would like to give fragments that contribute to good translations relatively high scores, and give fragments that contribute to bad translations relatively low scores, so that during decoding fragments that are known to contribute to good translations would be chosen over those that are known to contribute to bad translations. Furthermore, we would like to score each fragment in a derivation independently, since bad translations may contain good fragments, and vice-versa.

In practice, it is infeasible to rescore only those fragments seen during the rescoring process, due to the Goodman reduction for DOT. If we were to properly rescore each fragment, a new rule would need to be added to the grammar for each rule appearing in the fragment. Since the number of fragments is exponential, this would lead to a substantial increase in grammar size. Instead, we rescore the individual rules in the fragments, by evenly dividing the total amount of scoring mass among the rules of the particular fragment, and then assigning them the average of the rule scores over all fragments in which they appear. That is for each rule r in a fragment f consisting of $c_f(r)$ rules with score $\delta(f)$, the score of the rule is given as:

$$s(r) = \frac{\sum_{f:r \in f} \delta(f)/c_f(r)}{|f|} \quad (11)$$

This has the further advantage that we are allowing fragments that were unseen during tuning to be rescored according to previously seen fragment substructures.

To implement this scheme, we select a set of oracle translations for each sentence in the tuning data by evaluating all the candidate translations against the gold standard translation using the F-score (Turian et al., 2003), and selecting those with the highest F_1-measure, with exponent 1. We use GTM, rather than BLEU, because BLEU is not known to work well on a per-sentence level (Lavie et al., 2004) as needed for oracle selection. We then compare all the *target-side* fragments inferred in the translation process for each candidate translation against the fragments that yielded the oracles. There are two relevant parts of the fragments – the internal yields (i.e. the terminal leaves of the fragment) and the substitution sites (i.e. the frontiers where other fragments attach). We score the fragments rooted at the substitution sites separately from the parent fragment. We can uniquely identify the set of fragments that can be rooted at substitution sites by determining the span of the linked source-side derivation.

To compare two fragments, we define an edit distance between them. For a given fragment d, let $r(d)$ be the root of the fragment, let $r(d) \rightarrow rhs1$ be the left subtree of $r(d)$, and let $r(d) \rightarrow rhs2$ be the right subtree. The difference between a candidate fragment d_c and an oracle fragment d_{gs} is given by the equations in Table 1.

These equations define a minimum edit distance between two fragment trees, allowing subfragment order inversion, insertion, and deletion

$$\delta(d_c, d_{gs}) = \begin{cases} 0 & \text{if } d_c = d_{gs} \\ 1 & \text{if } d_c \neq d_{gs} \end{cases} \text{ Base case: } d_c \text{ and } d_{gs} \text{ are unary subtrees or substitution sites} \quad (9)$$

$$\delta(d_c, d_{gs}) = \min \begin{cases} \delta(d_c \rightarrow rhs1, d_{gs} \rightarrow rhs1) + \delta(d_c \rightarrow rhs2, d_{gs} \rightarrow rhs2), \\ \delta(d_c \rightarrow rhs2, d_{gs} \rightarrow rhs1) + \delta(d_c \rightarrow rhs1, d_{gs} \rightarrow rhs2) + 1, \\ \delta(d_c, d_{gs} \rightarrow rhs1) + |y(d_{gs} \rightarrow rhs2)|, \\ \delta(d_c, d_{gs} \rightarrow rhs2) + |y(d_{gs} \rightarrow rhs1)|, \\ \delta(d_c \rightarrow rhs1, d_{gs}) + |y(d_c \rightarrow rhs2)|, \\ \delta(d_c \rightarrow rhs2, d_{gs}) + |y(d_c \rightarrow rhs1)| \end{cases} \quad (10)$$

Table 1: The recursive relation defining the fragment difference between two fragments.

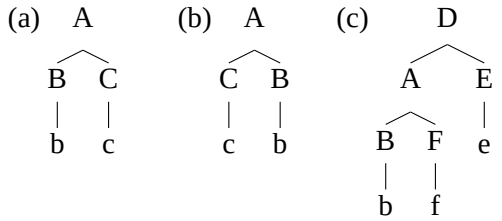


Figure 3: Comparing trees (a) and (b) with our distance metric yields a value of 1. The difference between trees (a) and (c) is 2, and for trees (b) and (c) the distance is 3.

as edit operations. For example, the only difference between trees (a) and (b) in Figure 3 is that their children have been inverted. To compare these trees using our distance metric, we first compute the first argument of the $\min$ function in Equation (10), directly comparing the structure of each immediate subtree. We then compute the second argument, obtaining the cost of performing an inversion, and finally compute the remaining arguments, assessing the cost of allowing each tree to be a direct subtree of the other. The result of this computation is 1, representing the inversion operation required to transform tree (a) into tree (b). If we compare trees (a) and (c) in Figure 3, we obtain a value of 2, given that the minimum operations required to transform tree (a) into tree (c) are inserting an additional subtree at the top level and then substituting the subtree rooted by C for the subtree rooted by F. If we compare tree (b) with tree (c) then the distance is 3, since we are now required to also replace the subtree rooted by C by the one rooted by B.

Since it is not efficient to compute the differences directly, we utilize common substructures and derive a dynamic programming implementation of the recursion. We compare each fragment against the set of oracle fragments for the same source span, and select the lowest cost as the score, assigning the candidate the negative difference between it and the oracle fragment it is most similar to, as in (12):

$$f(\langle d_s, d_t \rangle) = \max_{\langle d_s^o, d_t^o \rangle \in \mathbf{D}^o : d_s^o = d_s} -\delta(d_t, d_t^o) \quad (12)$$

In practice, given the Goodman reduction for DOT, we divide the fragment score by the number of rules in the fragment, and assign the average of those scores for each rule instance across all fragments rescored.

3.2 Normalized Structured Fragment Rescoring

In the structured fragment rescoring scheme, the scores that the fragments are assigned are the unnormalized edit distances between the two fragments. It may be better to normalize the fragment scores, rather than using the minimum number of tree transformations to convert one fragment into the other. We would expect that when comparing larger fragments, on average there would be more transformations needed to change one into the other than when comparing small fragments. However in the previous scheme, small fragments would have higher scores than large fragments, since fewer differences would be observed. The normalized score is given in (13):

$$f(\langle d_s, d_t \rangle) = \max_{\langle d_s^o, d_t^o \rangle \in \mathbf{D}^o : d_s^o = d_s} \log(1 - \delta(d_t, d_t^o)/ \max(|d_t|, |d_t^o|)) \quad (13)$$

Essentially, we are normalizing the edit distance by the maximum edit distance possible, namely the size of the largest fragment of the two being compared.

3.3 Fragment Surface Rescoring

The disadvantage of the minimum tree fragment edit approach is that it explicitly takes the internal

syntactic structure of the fragment into account. In comparing two fragments, they may have the same (or very similar) surface yields, but different internal structures. The previous approach would penalize the candidate fragment, even if its yield is quite close to the oracle. In this rescoring method, we extract the leaves of the candidate and oracle fragments, representing the substitution sites by the source span which their fragments cover. We then compare them using the Damerau-Levenshtein distance $\delta_{dl}(d_c, d_{gs})$ (Damerau, 1964) between the two fragment yields, and score them as in (14):

$$f(\langle d_s, d_t \rangle) = \max_{\langle d_s^o, d_t^o \rangle \in \mathbf{D}^o : d_s^o = d_s} -\delta_{dl}(d_t, d_t^o) \quad (14)$$

In Equation (14) we are selecting the maximal score for $\langle d_s, d_t \rangle$ from its comparison to all the possible corresponding oracle fragments. In this way, we are choosing to score $\langle d_s, d_t \rangle$ against the oracle fragment it is closest to.

4 Experiments

For our pilot experiments, we tested all the rescoring methods in the previous section on Spanish-to-English translation against the relative-frequency baseline. We randomly selected 10,000 sentences from the Europarl corpus (Koehn, 2005), and parsed and aligned the bitext as described in (Tinsley et al., 2009). From the parallel treebank, we extracted a Goodman reduction DOT grammar, as described in (Hearne, 2005), although on an order of magnitude greater amount of training data. Unlike (Bod, 2007), we did not use the unsupervised version of DOT, and did not attempt to scale up our amount of training data to his levels, although in ongoing work we are optimizing our system to be able to handle that amount of training data. To perform the rescoring, we randomly chose an additional 30K sentence pairs from the Spanish-to-English bitext. We rescored the grammar by translating the source side of the 10K training sentence pairs and 10K of the additional sentences, and using the methods in Section 3 to score the fragments derived in the translation process. We then performed the same experiment translating the full 40K-sentence set. Rules in the grammar that were not used during tuning were rescored using a default score defined to be the median of all scores observed.

Our system performs translation by first obtaining the n-best parses for the source sentences and then computing the k-best bilingual derivations for each source parse. In our experiments we used beams of $n = 10,000$ and $k = 5$. We also experimented with different values of α_0 and α_1 in Equation (7). We set these parameters manually, although in future work we will automatically tune them, perhaps using a MERT-like algorithm.

We tested our rescored grammars on a set of 2,000 randomly chosen Europarl sentences, and used a set of 200 randomly chosen sentences as a development test set. [1]

	BLEU	NIST	F-SCORE
Baseline	8.78	3.582	38.21

		2-8	4-6	5-5	6-4	8-2
BLEU	SFR	10.30	10.31	**10.32**	10.27	10.08
	NSFR	8.31	9.37	9.53	9.66	9.90
	FSR	10.19	10.25	10.18	10.19	9.93
NIST	SFR	3.792	3.805	**3.808**	3.800	3.781
	NSFR	3.431	3.638	3.661	3.693	3.722
	FSR	3.784	3.799	3.792	3.795	3.764
F-SCORE	SFR	**40.92**	40.82	40.86	40.84	40.78
	NSFR	37.53	39.50	39.93	40.38	40.78
	FSR	40.83	40.85	40.87	40.91	40.67

Table 2: Results on test set. Rescoring on 20K sentences. *SFR* stands for Structured Fragment Rescoring, *NSFR* for Normalized SFR and *FSR* for Fragment Surface Rescoring. *system-i-j* represents the corresponding system with $\alpha_0 = i$ and $\alpha_1 = j$. Underlined results are statistically significantly better than the baseline at $p = 0.01$.

	BLEU	NIST	F-SCORE
Baseline	8.78	3.582	38.21

		2-8	4-6	5-5	6-4	8-2
BLEU	SFR	**10.59**	10.58	10.41	10.38	10.08
	NSFR	8.61	9.71	9.90	9.96	9.93
	FSR	10.49	10.48	10.35	10.38	10.06
NIST	SFR	**3.841**	3.835	3.810	3.807	3.785
	NSFR	3.515	3.694	3.713	3.734	3.727
	FSR	3.834	3.833	3.820	3.816	3.784
F-SCORE	SFR	**41.12**	40.99	40.86	40.88	40.75
	NSFR	38.16	40.39	40.69	40.90	40.75
	FSR	41.03	41.02	41.01	40.98	40.72

Table 3: Results on test set. Rescoring on 40K sentences. Underlined are statistically significantly better than the baseline at $p = 0.01$.

5 Results

Translation quality results can be found in Tables 2 and 3. In these tables, columns labeled i-j indicate that the corresponding system was trained using parameters $\alpha_0 = i$ and $\alpha_1 = j$ in Equation 7. Statistical significance tests for NIST and BLEU were performed using Bootstrap Resampling (Koehn, 2004).

[1] All sentences, including the ones used for training, were limited to a length of at most 20 words.

	BLEU		NIST		F-SCORE	
Baseline	10.82		3.493		42.31	
		2-8	4-6	5-5	6-4	8-2
BLEU	SFR	11.34	**12.12**	11.94	11.97	11.78
	NSFR	9.68	10.99	11.38	11.63	11.30
	FSR	11.40	11.49	11.72	11.91	11.72
NIST	SFR	3.653	**3.727**	3.723	3.708	3.694
	NSFR	3.376	3.530	3.554	3.616	3.572
	FSR	3.655	3.675	3.698	3.701	3.675
F-SCORE	SFR	44.84	**45.47**	45.36	45.33	45.08
	NSFR	41.44	43.38	44.18	44.79	44.26
	FSR	44.68	44.91	45.15	45.19	44.82

Table 4: Results on development test set. Rescoring on 40K sentences.

As Table 2 indicates, all three rescoring methods significantly outperform the relative frequency baseline. The unnormalized structured fragment rescoring method performed the best, with the largest improvement of 1.5 BLEU points, a 17.5% relative improvement. We note that the BLEU scores for both the baseline and the experiments are low. This is to be expected, because the grammar is extracted from a very small bitext especially when the heterogeneity of the Europarl corpus is considered. In our analysis, only 32.5 percent of the test sentences had a complete source-side parse, meaning that a lot of structural information is lost contributing to arbitrary target-side ordering. In these experiments we did not use an additional language model. DOT (and many other syntax-based SMT systems) essentially have the target language model encoded within the translation model, since the inferences derived during translations link source structures to target structures, so in principle, no additional language model should be necessary. Furthermore, we only evaluate against a single reference, which also contributes to the lowering of absolute scores. To provide a sanity check against a state-of-the-art system, we trained the Moses phrase-based MT system (Koehn et al., 2007) using our training corpus, using no language model and using uniform feature weights, to provide a fair comparison against our baseline. We used this system to decode our development test set, and as a result we obtained a BLEU score of 10.72, which is comparable to the score obtained by our baseline on the same set.

When we scale up to tuning on 40,000 sentences we see an improvement in BLEU scores as well, as shown in Table 3. When tuning on 40K sentences, we observe an increase of 1.81 BLEU points on the best-performing system, which is a 20.6% improvement over the baseline. We note that rescoring on 20K sentences rescores approximately 275,000 rules out of 655,000 in the grammar, whereas rescoring on 40K sentences rescores approximately 280,000.

To analyze the benefits of the rescored grammar, we set aside a separate development set that we decoded with the grammar trained on 40K sentences. The results are presented in Table 4. The analysis is presented in Section 6.

Interestingly, there is a large difference between the normalized and unnormalized versions of the SFR scoring scheme. Our analysis suggests that the differences are mostly due to numerical issues, namely the difference in magnitude between the NSFR scores and the likelihood scores in the linear combination, and the default value assigned when the NSFR score was zero. In ongoing work, we are working to address these issues.

For most configurations the difference between SFR and FSR was not statistically significant at $p = 0.05$. Our analysis indicated that surface differences tended to co-occur with structural differences. We hypothesize that as we scale up to larger and more ambiguous grammars, the system will infer more derivations with the same yields, rendering a larger difference between the quality of the two scoring mechanisms.

6 Discussion

To analyze the advantages and disadvantages of our approach over the baseline, we closely examined and compared the derivations made on the devset translation by the SFR-scored grammar and the likelihood-scored grammar. Although the BLEU scores are rather low, there were several sentences in which the SFR-scored grammar showed a marked improvement over the baseline. We observed two types of improvements.

The first is where the rescored grammar gave us translations that, while still generally bad, were closer to the gold standard than the baseline translation. For example, the Spanish sentence “Y en tercer lugar , está el problema de la aplicación uniforme del Derecho comunitario .” translates into the gold standard “Thirdly , we have the problem of the uniform application of Community law .” The baseline grammar translates the sentence as “on third place , Transport and Tourism . I are the problems of the implementation standardised is the EU law .” with a GTM F-Score of 0.378,

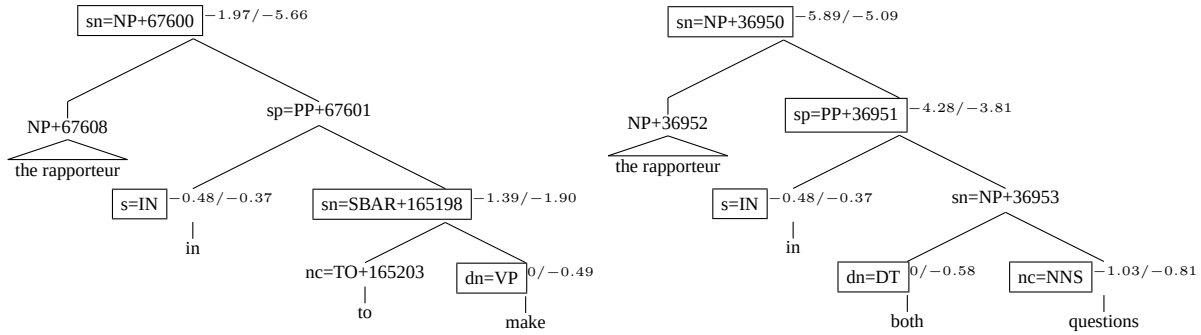


Figure 4: Target side of the highest-scoring translations for a sentence, according to the baseline system (left) and the SFR system (right). Boxed nodes are substitution sites. Scores in superscripts denote the score of the sub-derivation according to the baseline and to the SFR system.

and the rescored grammar outputs the translation "to there in the third place , I are the problem of the implementation standardised is the Community law .", with an F-Score of 0.5. While many of the fragments in the derivations that yielded these two translations differ, the ones we would like to focus on are the fragments that yield the translation of "comunitario". The grammar contains several competing unary fragment pairs for "comunitaro". In the baseline grammar, the pair (`aq=NNP` → `comunitario`, `aq=NNP` → `EU`) has a score of -0.693147, whereas the pair (`aq=NNP` → `comunitario`, `aq=NNP` → `Community`) has a score of -1.38629. In the rescored grammar however, (`aq=NNP` → `comunitario`, `aq=NNP` → `EU`) has a score of -0.762973, whereas (`aq=NNP` → `comunitario`, `aq=NNP` → `Community`) has a score of -0.74399. In effect, the rescoring scheme rescored the word alignment itself. This suggests that in future work, it may be possible to integrate a word aligner or fragment aligner directly into the MT training method.

The other improvement was where the baseline and the SFR-scored grammar output translations of roughly the same quality according to the evaluation measure, yet in terms of human evaluation, the SFR translation was much better than the baseline translation. For instance, our devset contained the Spanish sentence "Estoy de acuerdo con el ponente en dos cuestiones ." The baseline translation given is "I agree with the rapporteur in to make .", and the SFR-scored translation given is "I agree with the rapporteur in both questions .". While both translations have the same GTM score against the gold standard "I agree with the rapporteur on two issues .", clearly, the second one is of far higher quality than the first. As we can see in Figure 4, the derivation over the substring "in both questions" gets a higher score than "in to make" when translated with the rescored grammar. In the baseline, "en dos cuestiones" is not translated as a whole unit – rather, the derivation of "el ponente en dos cuestiones" is decomposed into four subderivations, yielding "el" "ponente" "en" "dos cuestiones", where each of those is translated separately, into "∅" "the rapporteur" "in" and "to make". The SFR-scored grammar, however, outputs a different bilingual derivation. The source is decomposed into five sub-derivations, one for each word, and each word is translated separately. Then, the rescored target fragments set the proper target-side word order and select the target-side words that maximize the score of the subderivation covering the source span. We note that in this example, the score of translating "dos" to "make" was higher than the score of translating "dos" to "both". However, the higher level target fragment that composed the translation of "dos" together with the translation of "cuestiones" yielded a higher score when composing "both questions" rather than "to make".

7 Conclusions and Future Work

The results presented above indicate that augmenting the scoring mechanism with an accuracy-based measure is a promising direction for translation quality improvement. It gives us a statistically significant improvement over the baseline, and our analysis has indicated that the system is indeed making better decisions, moving us a step closer towards the goal of making translation decisions based on the hypothesis of the resulting transla-

tion's accuracy.

Now that we have demonstrated that translation quality can be improved by incorporating a measure of fragment quality into the scoring scheme, our immediate next step is to optimize our system so that we can scale up to significantly larger training and tuning sets, and determine whether the improvements we have noted carry over when the likelihood is computed from more data. Afterwards, we will implement a training scheme to maximize an accuracy-based objective function, for instance, by minimizing the difference between the scores of the highest-scoring derivation and the oracle derivations, in effect maximizing the score of the highest-scoring translation.

The rescoring method presented in this paper need not be limited to DOT. Fragments can be thought of as analogous to phrases in Phrase-Based SMT systems – we could implement a similar rescoring system for phrase-based systems, where we generate several candidate translations for source sentences in a tuning set, and score each phrase used against the phrases used in a set of oracles. More broadly, we could potentially take any statistical MT system, and compare the features of all candidates generated against those of oracle translations, and score those that are closer to the oracle higher than those further away.

Finally, by explicitly framing the translation problem as a search problem, where we are divorcing the inferences in the search space (i.e. the model) from the path we take to find the optimal inference according to some criterion (i.e. the scoring scheme), we can remove some of the variability when comparing two models or scoring mechanisms (Lopez, 2009).

Acknowledgements

This work is supported by Science Foundation Ireland (Grant No. 07/CE/I1142). We would like to thank the anonymous reviewers for their helpful comments and suggestions.

References

R. Bod. 2007. Unsupervised syntax-based machine translation: The contribution of discontiguous phrases. In *Proceedings of the 11th Machine Translation Summit*, pages 51–57, Copenhagen, Denmark.

P. F. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

F. J. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7(3):171–176.

J. Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), Companion Volume*, pages 205–208, Sapporo.

M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia.

J. Goodman. 1996. Efficient algorithms for parsing the DOP model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 143–152, Philadelphia, PA.

M. Hearne and A. Way. 2003. Seeing the wood for the trees: Data-oriented translation. In *Proceedings of the Ninth Machine Translation Summit*, pages 165–172, New Orleans, LA.

M. Hearne and A. Way. 2006. Disambiguation strategies for data-oriented translation. In *Proceedings of the 11th Conference of the European Association for Machine Translation*, pages 59–68, Oslo, Norway.

M. Hearne. 2005. *Data-Oriented Models of Parsing and Translation*. Ph.D. thesis, Dublin City University, Dublin, Ireland.

M. Johnson. 2002. The DOP estimation method is biased and inconsistent. *Computational Linguistics*, 28(1):71–76, March.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, demonstation session*, pages 177–180, Prague, Czech Republic.

P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of*

the Conference on Empirical Methods in Natural Language Processing, pages 388–395, Barcelona, Spain.

P. Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Machine Translation Summit X*, pages 79–86, Phuket, Thailand.

A. Lavie, K. Sagae, and S. Jayaraman. 2004. The significance of recall in automatic metrics for MT evaluation. In *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas*, pages 134–143, Washington, DC.

A. Lopez. 2009. Translation as weighted deduction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 532–540, Athens, Greece.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

A. Poutsma. 2000. Data-oriented translation. In *The 18th International Conference on Computational Linguistics*, pages 635–641, Saarbrücken, Germany.

C. Tillmann and T. Zhang. 2006. A discriminative global training algorithm for statistical MT. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 721–728, Sydney, Australia.

J. Tinsley, M. Hearne, and A. Way. 2009. Parallel treebanks in phrase-based statistical machine translation. In *Proceedings of the Tenth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 318–331, Mexico City, Mexico.

J. Turian, L. Shen, and I. D. Melamed. 2003. Evaluation of machine translation and its evaluation. In *Proceedings of the Ninth Machine Translation Summit*, pages 386–393, New Orleans, LA.

D. Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–404.

K. Yamada and K. Knight. 2001. A syntax-based statistical translation model. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530, Toulouse, France.

K. Yamada and K. Knight. 2002. A decoder for syntax-based statistical MT. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 303–310, Philadelphia, PA.

B. Zhao and S. Chen. 2009. A simplex armijo downhill algorithm for optimizing statistical machine translation decoding parameters. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 21–24, Boulder, Colorado.