Consistent Translation using Discriminative Learning: A Translation Memory-inspired Approach[*]

Yanjun Ma[†] **Yifan He**[‡] **Andy Way**[‡] **Josef van Genabith**[‡]
[†] Baidu Inc., Beijing, China
yma@baidu.com
[‡]Centre for Next Generation Localisation
School of Computing, Dublin City University
{yhe,away,josef}@computing.dcu.ie

Abstract

We present a discriminative learning method to improve the consistency of translations in phrase-based Statistical Machine Translation (SMT) systems. Our method is inspired by Translation Memory (TM) systems which are widely used by human translators in industrial settings. We constrain the translation of an input sentence using the most similar 'translation example' retrieved from the TM. Differently from previous research which used simple fuzzy match thresholds, these constraints are imposed using discriminative learning to optimise the translation performance. We observe that using this method can benefit the SMT system by not only producing consistent translations, but also improved translation outputs. We report a 0.9 point improvement in terms of BLEU score on English–Chinese technical documents.

1 Introduction

Translation consistency is an important factor for large-scale translation, especially for domain-specific translations in an industrial environment. For example, in the translation of technical documents, lexical as well as structural consistency is essential to produce a fluent target-language sentence. Moreover, even in the case of translation errors, consistency in the errors (e.g. repetitive error patterns) are easier to diagnose and subsequently correct by translators.

In phrase-based SMT, translation models and language models are automatically learned and/or generalised from the training data, and a translation is produced by maximising a weighted combination of these models. Given that global contextual information is not normally incorporated, and that training data is usually noisy in nature, there is no guarantee that an SMT system can produce translations in a consistent manner.

On the other hand, TM systems – widely used by translators in industrial environments for enterprise localisation by translators – can shed some light on mitigating this limitation. TM systems can assist translators by retrieving and displaying previously translated similar 'example' sentences (displayed as source-target pairs, widely called 'fuzzy matches' in the localisation industry (Sikes, 2007)). In TM systems, fuzzy matches are retrieved by calculating the similarity or the so-called 'fuzzy match score' (ranging from 0 to 1 with 0 indicating no matches and 1 indicating a full match) between the input sentence and sentences in the source side of the translation memory.

When presented with fuzzy matches, translators can then avail of useful chunks in previous translations while composing the translation of a new sentence. Most translators only consider a few sentences that are most similar to the current input sentence; this process can inherently improve the consistency of translation, given that the new translations produced by translators are likely to be similar to the target side of the fuzzy match they have consulted.

Previous research as discussed in detail in Sec-

[*] This work was done while the first author was in the Centre for Next Generation Localisation at Dublin City University.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1239–1248,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

tion 2 has focused on using fuzzy match score as a threshold when using the target side of the fuzzy matches to constrain the translation of the input sentence. In our approach, we use a more fine-grained discriminative learning method to determine whether the target side of the fuzzy matches should be used as a constraint in translating the input sentence. We demonstrate that our method can consistently improve translation quality.

The rest of the paper is organized as follows: we begin by briefly introducing related research in Section 2. We present our discriminative learning method for consistent translation in Section 3 and our feature design in Section 4. We report the experimental results in Section 5 and conclude the paper and point out avenues for future research in Section 6.

2 Related Research

Despite the fact that TM and MT integration has long existed as a major challenge in the localisation industry, it has only recently received attention in main-stream MT research. One can loosely combine TM and MT at sentence (called segments in TMs) level by choosing one of them (or both) to recommend to the translators using automatic classifiers (He et al., 2010), or simply using fuzzy match score or MT confidence measures (Specia et al., 2009).

One can also tightly integrate TM with MT at the sub-sentence level. The basic idea is as follows: given a source sentence to translate, we firstly use a TM system to retrieve the most similar 'example' source sentences together with their translations. If matched chunks between input sentence and fuzzy matches can be detected, we can directly re-use the corresponding parts of the translation in the fuzzy matches, and use an MT system to translate the remaining chunks.

As a matter of fact, implementing this idea is pretty straightforward: a TM system can easily detect the word alignment between the input sentence and the source side of the fuzzy match by retracing the paths used in calculating the fuzzy match score. To obtain the translation for the matched chunks, we just require the word alignment between source and target TM matches, which can be addressed using state-of-the-art word alignment techniques. More importantly, albeit not explicitly spelled out in previous work, this method can potentially increase the consistency of translation, as the translation of new input sentences is closely informed and guided (or constrained) by previously translated sentences.

There are several different ways of using the translation information derived from fuzzy matches, with the following two being the most widely adopted: 1) to add these translations into a phrase table as in (Biçici and Dymetman, 2008; Simard and Isabelle, 2009), or 2) to mark up the input sentence using the relevant chunk translations in the fuzzy match, and to use an MT system to translate the parts that are not marked up, as in (Smith and Clark, 2009; Koehn and Senellart, 2010; Zhechev and van Genabith, 2010). It is worth mentioning that translation consistency was not explicitly regarded as their primary motivation in this previous work. Our research follows the direction of the second strand given that consistency can no longer be guaranteed by constructing another phrase table.

However, to categorically reuse the translations of matched chunks without any differentiation could generate inferior translations given the fact that the context of these matched chunks in the input sentence could be completely different from the source side of the fuzzy match. To address this problem, both (Koehn and Senellart, 2010) and (Zhechev and van Genabith, 2010) used fuzzy match score as a threshold to determine whether to reuse the translations of the matched chunks. For example, (Koehn and Senellart, 2010) showed that reusing these translations as large rules in a hierarchical system (Chiang, 2005) can be beneficial when the fuzzy match score is above 70%, while (Zhechev and van Genabith, 2010) reported that it is only beneficial to a phrase-based system when the fuzzy match score is above 90%.

Despite being an informative measure, using fuzzy match score as a threshold has a number of limitations. Given the fact that fuzzy match score is normally calculated based on Edit Distance (Levenshtein, 1966), a low score does not necessarily imply that the fuzzy match is harmful when used to constrain an input sentence. For example, in longer sentences where fuzzy match scores tend to be low, some chunks and the corresponding translations within the sentences can still be useful. On the other hand, a high score cannot fully guarantee the usefulness of a particular translation. We address this problem using discriminative learning.

3 Constrained Translation with Discriminative Learning

3.1 Formulation of the Problem

Given a sentence $\mathbf{e}$ to translate, we retrieve the most similar sentence $\mathbf{e}'$ from the translation memory associated with target translation $\mathbf{f}'$. The m common "phrases" $\mathbf{\bar{e}_1^m}$ between $\mathbf{e}$ and $\mathbf{e}'$ can be identified. Given the word alignment information between $\mathbf{e}'$ and $\mathbf{f}'$, one can easily obtain the corresponding translations $\mathbf{\bar{f}'{}_1^m}$ for each of the phrases in $\mathbf{\bar{e}_1^m}$. This process can derive a number of "phrase pairs" $< \bar{e}_m, \bar{f}'_m >$, which can be used to specify the translations of the matched phrases in the input sentence. The remaining words without specified translations will be translated by an MT system.

For example, given an input sentence $e_1 e_2 \cdots e_i e_{i+1} \cdots e_I$, and a phrase pair $< \bar{e}, \bar{f}' >$, $\bar{e} = e_i e_{i+1}$, $\bar{f}' = f'_j f'_{j+1}$ derived from the fuzzy match, we can mark up the input sentence as:

$e_1 e_2 \cdots$ <tm="$f'_j f'_{j+1}$"> $e_i e_{i+1}$ < /tm> $\cdots e_I$.

Our method to constrain the translations using TM fuzzy matches is similar to (Koehn and Senellart, 2010), except that the word alignment between $\mathbf{e}'$ and $\mathbf{f}'$ is the intersection of bidirectional GIZA++ (Och and Ney, 2003) posterior alignments. We use the intersected word alignment to minimise the noise introduced by word alignment of only one direction in marking up the input sentence.

3.2 Discriminative Learning

Whether the translation information from the fuzzy matches should be used or not (i.e. whether the input sentence should be marked up) is determined using a discriminative learning procedure. The translation information refers to the "phrase pairs" derived using the method described in Section 3.1. We cast this problem as a binary classification problem.

3.2.1 Support Vector Machines

SVMs (Cortes and Vapnik, 1995) are binary classifiers that classify an input instance based on decision rules which minimise the regularised error function in (1):

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{s. t.} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geqslant 1 - \xi_i \\ & \xi_i \geqslant 0 \end{aligned} \tag{1}$$

where $(\mathbf{x}_i, y_i) \in R^n \times \{+1, -1\}$ are l training instances that are mapped by the function ϕ to a higher dimensional space. $\mathbf{w}$ is the weight vector, ξ is the relaxation variable and $C > 0$ is the penalty parameter.

Solving SVMs is viable using a kernel function K in (1) with $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j)$. We perform our experiments with the Radial Basis Function (RBF) kernel, as in (2):

$$K(\mathbf{x}_i, \mathbf{x}_j) = exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2), \gamma > 0 \tag{2}$$

When using SVMs with the RBF kernel, we have two free parameters to tune on: the cost parameter C in (1) and the radius parameter γ in (2).

In each of our experimental settings, the parameters C and γ are optimised by a brute-force grid search. The classification result of each set of parameters is evaluated by cross validation on the training set.

The SVM classifier will thus be able to predict the usefulness of the TM fuzzy match, and determine whether the input sentence should be marked up using relevant phrase pairs derived from the fuzzy match before sending it to the SMT system for translation. The classifier uses features such as the fuzzy match score, the phrase and lexical translation probabilities of these relevant phrase pairs, and additional syntactic dependency features. Ideally the classifier will decide to mark up the input sentence if the translations of the marked phrases are accurate when taken contextual information into account. As large-scale manually annotated data is not available for this task, we use automatic TER scores (Snover et al., 2006) as the measure for training data annotation.

We label the training examples as in (3):

$$y = \begin{cases} +1 & \text{if } TER(\text{w. markup}) < TER(\text{w/o markup}) \\ -1 & \text{if } TER(\text{w/o markup}) \geq TER(\text{w. markup}) \end{cases} \tag{3}$$

Each instance is associated with a set of features which are discussed in more detail in Section 4.

3.2.2 Classification Confidence Estimation

We use the techniques proposed by (Platt, 1999) and improved by (Lin et al., 2007) to convert classification margin to posterior probability, so that we can easily threshold our classifier (cf. Section 5.4.2).

Platt's method estimates the posterior probability with a sigmoid function, as in (4):

$$Pr(y=1|\mathbf{x}) \approx P_{A,B}(f) \equiv \frac{1}{1+exp(Af+B)} \quad (4)$$

where $f = f(\mathbf{x})$ is the decision function of the estimated SVM. A and B are parameters that minimise the cross-entropy error function F on the training data, as in (5):

$$\min_{z=(A,B)} F(z) = -\sum_{i=1}^{l}(t_i log(p_i) + (1-t_i)log(1-p_i)),$$
$$\text{where } p_i = P_{A,B}(f_i), \text{and } t_i = \begin{cases} \frac{N_+ +1}{N_+ +2} & \text{if } y_i = +1 \\ \frac{1}{N_- +2} & \text{if } y_i = -1 \end{cases} \quad (5)$$

where $z = (A,B)$ is a parameter setting, and N_+ and N_- are the numbers of observed positive and negative examples, respectively, for the label y_i. These numbers are obtained using an internal cross-validation on the training set.

4 Feature Set

The features used to train the discriminative classifier, all on the sentence level, are described in the following sections.

4.1 The TM Feature

The TM feature is the fuzzy match score, which indicates the overall similarity between the input sentence and the source side of the TM output. If the input sentence is similar to the source side of the matching segment, it is more likely that the matching segment can be used to mark up the input sentence.

The calculation of the fuzzy match score itself is one of the core technologies in TM systems, and varies among different vendors. We compute fuzzy match cost as the minimum Edit Distance (Levenshtein, 1966) between the source and TM entry, normalised by the length of the source as in (6), as most of the current implementations are based on edit distance while allowing some additional flexible matching.

$$h_{fm}(\mathbf{e}) = \min_{\mathbf{s}} \frac{EditDistance(\mathbf{e}, \mathbf{s})}{Len(\mathbf{e})} \quad (6)$$

where $\mathbf{e}$ is the sentence to translate, and $\mathbf{s}$ is the source side of an entry in the TM. For fuzzy match scores F, h_{fm} roughly corresponds to $1 - F$.

4.2 Translation Features

We use four features related to translation probabilities, i.e. the phrase translation and lexical probabilities for the phrase pairs $< \bar{e}_m, \bar{f'}_m >$ derived using the method in Section 3.1. Specifically, we use the phrase translation probabilities $p(\bar{f'}_m|\bar{e}_m)$ and $p(\bar{e}_m|\bar{f'}_m)$, as well as the lexical translation probabilities $p_{lex}(\bar{f'}_m|\bar{e}_m)$ and $p_{lex}(\bar{e}_m|\bar{f'}_m)$ as calculated in (Koehn et al., 2003). In cases where multiple phrase pairs are used to mark up one single input sentence $\mathbf{e}$, we use a unified score for each of the four features, which is an average over the corresponding feature in each phrase pair. The intuition behind these features is as follows: phrase pairs $< \bar{e}_m, \bar{f'}_m >$ derived from the fuzzy match should also be reliable with respect to statistically produced models.

We also have a count feature, i.e. the number of phrases used to mark up the input sentence, and a binary feature, i.e. whether the phrase table contains at least one phrase pair $< \bar{e}_m, \bar{f'}_m >$ that is used to mark up the input sentence.

4.3 Dependency Features

Given the phrase pairs $< \bar{e}_m, \bar{f'}_m >$ derived from the fuzzy match, and used to translate the corresponding chunks of the input sentence (cf. Section 3.1), these translations are more likely to be coherent in the context of the particular input sentence if the matched parts on the input side are syntactically and semantically related.

For matched phrases $\bar{e}_m$ between the input sentence and the source side of the fuzzy match, we define the contextual information of the input side using dependency relations between words e_m in $\bar{e}_m$ and the remaining words e_j in the input sentence $\mathbf{e}$.

We use the Stanford parser to obtain the dependency structure of the input sentence. We add a pseudo-label SYS_PUNCT to punctuation marks, whose governor and dependent are both the punctuation mark. The dependency features designed to capture the context of the matched input phrases $\bar{e}_m$ are as follows:

Coverage features measure the coverage of dependency labels on the input sentence in order to obtain a bigger picture of the matched parts in the input. For each dependency label *L*, we consider its head or modifier as *covered* if the corresponding input word e_m is covered by a matched phrase $\bar{e}_m$. Our coverage features are the frequencies of governor and dependent coverage calculated separately for each dependency label.

Position features identify whether the head and the tail of a sentence are matched, as these are the cases in which the matched translation is not affected by the preceding words (when it is the head) or following words (when it is the tail), and is therefore more reliable. The feature is set to 1 if this happens, and to 0 otherwise. We distinguish among the possible dependency labels, the head or the tail of the sentence, and whether the aligned word is the governor or the dependent. As a result, each permutation of these possibilities constitutes a distinct binary feature.

The consistency feature is a single feature which determines whether matched phrases $\bar{e}_m$ belong to a consistent dependency structure, instead of being distributed discontinuously around in the input sentence. We assume that a consistent structure is less influenced by its surrounding context. We set this feature to 1 if every word in $\bar{e}_m$ is dependent on another word in $\bar{e}_m$, and to 0 otherwise.

5 Experiments

5.1 Experimental Setup

Our data set is an English–Chinese translation memory with technical translation from Symantec, consisting of 87K sentence pairs. The average sentence length of the English training set is 13.3 words and the size of the training set is comparable to the larger TMs used in the industry. Detailed corpus statistics about the training, development and test sets for the SMT system are shown in Table 1.

	Train	Develop	Test
SENTENCES	86,602	762	943
ENG. TOKENS	1,148,126	13,955	20,786
ENG. VOC.	13,074	3,212	3,115
CHI. TOKENS	1,171,322	10,791	16,375
CHI. VOC.	12,823	3,212	1,431

Table 1: Corpus Statistics

Scores	Sentences	Words	W/S
(0.9, 1.0)	80	1526	19.0750
(0.8, 0.9]	96	1430	14.8958
(0.7, 0.8]	110	1596	14.5091
(0.6, 0.7]	74	1031	13.9324
(0.5, 0.6]	104	1811	17.4135
(0, 0.5]	479	8972	18.7307

Table 2: Composition of test subsets based on fuzzy match scores

The composition of test subsets based on fuzzy match scores is shown in Table 2. We can see that sentences in the test sets are longer than those in the training data, implying a relatively difficult translation task. We train the SVM classifier using the libSVM (Chang and Lin, 2001) toolkit. The SVM-training and validation is on the same training sentences[1] as the SMT system with 5-fold cross validation.

The SVM hyper-parameters are tuned using the training data of the first fold in the 5-fold cross validation via a brute force grid search. More specifically, for parameter C in (1), we search in the range $[2^{-5}, 2^{15}]$, while for parameter γ (2) we search in the range $[2^{-15}, 2^{3}]$. The step size is 2 on the exponent.

We conducted experiments using a standard log-linear PB-SMT model: GIZA++ implementation of IBM word alignment model 4 (Och and Ney, 2003), the refinement and phrase-extraction heuristics described in (Koehn et al., 2003), minimum-error-rate training (Och, 2003), a 5-gram language model with Kneser-Ney smoothing (Kneser and Ney, 1995) trained with SRILM (Stolcke, 2002) on the Chinese side of the training data, and Moses (Koehn et al., 2007) which is capable of handling user-specified translations for some portions of the input during decoding. The maximum phrase length is set to 7.

5.2 Evaluation

The performance of the phrase-based SMT system is measured by BLEU score (Papineni et al., 2002) and TER (Snover et al., 2006). Significance test-

[1] We have around 87K sentence pairs in our training data. However, for 67.5% of the input sentences, our MT system produces the same translation irrespective of whether the input sentence is marked up or not.

ing is carried out using approximate randomisation (Noreen, 1989) with a 95% confidence level.

We also measure the quality of the classification by precision and recall. Let A be the set of predicted markup input sentences, and B be the set of input sentences where the markup version has a lower TER score than the plain version. We standardly define precision P and recall R as in (7):

$$P = \frac{|A \bigcap B|}{|A|}, R = \frac{|A \bigcap B|}{|B|} \qquad (7)$$

5.3 Cross-fold translation

In order to obtain training samples for the classifier, we need to label each sentence in the SMT training data as to whether marking up the sentence can produce better translations. To achieve this, we translate both the marked-up versions and plain versions of the sentence and compare the two translations using the sentence-level evaluation metric TER.

We do not make use of additional training data to translate the sentences for SMT training, but instead use cross-fold translation. We create a new training corpus T by keeping 95% of the sentences in the original training corpus, and creating a new test corpus H by using the remaining 5% of the sentences. Using this scheme we make 20 different pairs of corpora (T_i, H_i) in such a way that each sentence from the original training corpus is in exactly one H_i for some $1 \leq i \leq 20$. We train 20 different systems using each T_i, and use each system to translate the corresponding H_i as well as the marked-up version of H_i using the procedure described in Section 3.1. The development set is kept the same for all systems.

5.4 Experimental Results

5.4.1 Translation Results

Table 3 contains the translation results of the SMT system when we use discriminative learning to mark up the input sentence (MARKUP-DL). The first row (BASELINE) is the result of translating plain test sets without any markup, while the second row is the result when all the test sentences are marked up. We also report the oracle scores, i.e. the upperbound of using our discriminative learning approach. As we can see from this table, we obtain significantly inferior results compared to the the Baseline system if we categorically mark up all the input sentences using phrase pairs derived from fuzzy matches. This is reflected by an absolute 1.4 point drop in BLEU score and a 1.8 point increase in TER. On the other hand, both the oracle BLEU and TER scores represent as much as a 2.5 point improvement over the baseline. Our discriminative learning method (MARKUP-DL), which automatically classifies whether an input sentence should be marked up, leads to an increase of 0.7 absolute BLEU points over the BASELINE, which is statistically significant. We also observe a slight decrease in TER compared to the BASELINE. Despite there being much room for further improvement when compared to the Oracle score, the discriminative learning method appears to be effective not only in maintaining translation consistency, but also a statistically significant improvement in translation quality.

	TER	BLEU
BASELINE	39.82	45.80
MARKUP	41.62	44.41
MARKUP-DL	39.61	46.46
ORACLE	37.27	48.32

Table 3: Performance of Discriminative Learning (%)

5.4.2 Classification Confidence Thresholding

To further analyse our discriminative learning approach, we report the classification results on the test set using the SVM classifier. We also investigate the use of classification confidence, as described in Section 3.2.2, as a threshold to boost classification precision if required. Table 4 shows the classification and translation results when we use different confidence thresholds. The default classification confidence is 0.50, and the corresponding translation results were described in Section 5.4.1. We investigate the impact of increasing classification confidence on the performance of the classifier and the translation results. As can be seen from Table 4, increasing the classification confidence up to 0.70 leads to a steady increase in classification precision with a corresponding sacrifice in recall. The fluctuation in classification performance has an impact on the translation results as measured by BLEU and TER. We can see that the best BLEU as well as TER scores are achieved when we set the classification confidence to 0.60, representing a modest improve-

	Classification Confidence						
	0.50	0.55	0.60	0.65	0.70	0.75	0.80
BLEU	46.46	46.65	**46.69**	46.59	46.34	46.06	46.00
TER	39.61	39.46	**39.32**	39.36	39.52	39.71	39.71
P	60.00	68.67	70.31	**74.47**	72.97	64.28	88.89
R	32.14	29.08	22.96	**17.86**	13.78	9.18	4.08

Table 4: The impact of classification confidence thresholding

ment over the default setting (0.50). Despite the higher precision when the confidence is set to 0.7, the dramatic decrease in recall cannot be compensated for by the increase in precision.

We can also observe from Table 4 that the recall is quite low across the board, and the classification results become unstable when we further increase the level of confidence to above 0.70. This indicates the degree of difficulty of this classification task, and suggests some directions for future research as discussed at the end of this paper.

5.4.3 Comparison with Previous Work

As discussed in Section 2, both (Koehn and Senellart, 2010) and (Zhechev and van Genabith, 2010) used fuzzy match score to determine whether the input sentences should be marked up. The input sentences are only marked up when the fuzzy match score is above a certain threshold. We present the results using this method in Table 5. From this table, we can see an inferior performance compared to the BASELINE results (cf. Table 3) when the fuzzy match score is below 0.70. A modest gain can only be achieved when the fuzzy match score is above 0.8. This is slightly different from the conclusions drawn in (Koehn and Senellart, 2010), where gains are observed when the fuzzy match score is above 0.7, and in (Zhechev and van Genabith, 2010) where gains are only observed when the score is above 0.9. Comparing Table 5 with Table 4, we can see that our classification method is more effective. This confirms our argument in the last paragraph of Section 2, namely that fuzzy match score is not informative enough to determine the usefulness of the subsentences in a fuzzy match, and that a more comprehensive set of features, as we have explored in this paper, is essential for the discriminative learning-based method to work.

	Fuzzy Match Scores				
	0.50	0.60	0.70	0.80	0.90
BLEU	45.13	45.55	45.58	45.84	45.82
TER	40.99	40.62	40.56	40.29	40.07

Table 5: Performance using fuzzy match score for classification

FM Scores	w. markup	w/o markup
[0,0.5]	37.75	62.24
(0.5,0.6]	40.64	59.36
(0.6,0.7]	40.94	59.06
(0.7,0.8]	46.67	53.33
(0.8,0.9]	54.28	45.72
(0.9,1.0]	44.14	55.86

Table 6: Percentage of training sentences with markup vs without markup grouped by fuzzy match (FM) score ranges

To further validate our assumption, we analyse the training sentences by grouping them according to their fuzzy match score ranges. For each group of sentences, we calculate the percentage of sentences where markup (and respectively without markup) can produce better translations. The statistics are shown in Table 6. We can see that for sentences with fuzzy match scores lower than 0.8, more sentences can be better translated without markup. For sentences where fuzzy match scores are within the range $(0.8, 0.9]$, more sentences can be better translated with markup. However, within the range $(0.9, 1.0]$, surprisingly, actually more sentences receive better translation without markup. This indicates that fuzzy match score is not a good measure to predict whether fuzzy matches are beneficial when used to constrain the translation of an input sentence.

5.5 Contribution of Features

We also investigated the contribution of our different feature sets. We are especially interested in the contribution of dependency features, as they re-

	Example 1
w/o markup	after policy name , type the name of the policy (it shows new host integrity policy by default) .
Translation	在 “ 策略 ” 名称 后面 ， 键入 策略 的 名称 (名称 显示 为 “ 新 主机 完整性 策略 默认 ） 。
w. markup	after policy name <tm translation=“， 键入 策略 名称 （ 默认 显示 “ 新 主机 完整性 策略 ” ） 。”>, type the name of the policy (it shows new host integrity policy by default) .< /tm>
Translation	在 “ 策略 ” 名称 后面 ， 键入 策略 名称 （ 默认 显示 “ 新 主机 完整性 策略 ” ） 。
Reference	在 “ 策略 名称 ” 后面 ， 键入 策略 名称 （ 默认 显示 “ 新 主机 完整性 策略 ” ） 。
	Example 2
w/o markup	changes apply only to the specific scan that you select .
Translation	更改 仅 适用于 特定 扫描 的 规则 。
w. markup	changes apply only to the specific scan that you select <tm translation=“。”>.< /tm>
Translation	更改 仅 适用于 您 选择 的 特定 扫描 。
Reference	更改 只 应用于 您 选择 的 特定 扫描 。

flect whether translation consistency can be captured using syntactic knowledge. The classification and translation results using different features are reported in Table 7. We observe a significant improvement in both classification precision and recall by adding dependency (DEP) features on top of TM and translation features. As a result, the translation quality also significantly improves. This indicates that dependency features which can capture structural and semantic similarities are effective in gauging the usefulness of the phrase pairs derived from the fuzzy matches. Note also that without including the dependency features, our discriminative learning method cannot outperform the BASELINE (cf. Table 3) in terms of translation quality.

	TER	BLEU	P	R
TM+TRANS	40.57	45.51	52.48	27.04
+DEP	39.61	46.46	60.00	32.14

Table 7: Contribution of Features (%)

5.6 Improved Translations

In order to pinpoint the sources of improvements by marking up the input sentence, we performed some manual analysis of the output. We observe that the improvements can broadly be attributed to two reasons: 1) the use of long phrase pairs which are missing in the phrase table, and 2) deterministically using highly reliable phrase pairs.

Phrase-based SMT systems normally impose a limit on the length of phrase pairs for storage and speed considerations. Our method can overcome this limitation by retrieving and reusing long phrase pairs on the fly. A similar idea, albeit from a different perspective, was explored by (Lopez, 2008), where he proposed to construct a phrase table on the fly for each sentence to be translated. Differently from his approach, our method directly translates part of the input sentence using fuzzy matches retrieved on the fly, with the rest of the sentence translated by the pre-trained MT system. We offer some more insights into the advantages of our method by means of a few examples.

Example 1 shows translation improvements by using long phrase pairs. Compared to the reference translation, we can see that for the underlined phrase, the translation without markup contains (i) word ordering errors and (ii) a missing right quotation mark. In Example 2, by specifying the translation of the final punctuation mark, the system correctly translates the relative clause ‘that you select’. The translation of this relative clause is missing when translating the input without markup. This improvement can be partly attributed to the reduction in search errors by specifying the highly reliable translations for phrases in an input sentence.

6 Conclusions and Future Work

In this paper, we introduced a discriminative learning method to tightly integrate fuzzy matches retrieved using translation memory technologies with phrase-based SMT systems to improve translation consistency. We used an SVM classifier to predict whether phrase pairs derived from fuzzy matches could be used to constrain the translation of an input sentence. A number of feature functions including a series of novel dependency features were used to train the classifier. Experiments demonstrated that discriminative learning is effective in improving translation quality and is more informative than the fuzzy match score used in previous research. We report a statistically significant 0.9 absolute improvement in BLEU score using a procedure to promote translation consistency.

As mentioned in Section 2, the potential improvement in sentence-level translation consistency using our method can be attributed to the fact that the translation of new input sentences is closely informed and guided (or constrained) by previously translated sentences using global features such as dependencies. However, it is worth noting that the level of gains in translation consistency is also dependent on the nature of the TM itself; a self-contained coherent TM would facilitate consistent translations. In the future, we plan to investigate the impact of TM quality on translation consistency when using our approach. Furthermore, we will explore methods to promote translation consistency at document level.

Moreover, we also plan to experiment with phrase-by-phrase classification instead of sentence-by-sentence classification presented in this paper, in order to obtain more stable classification results. We also plan to label the training examples using other sentence-level evaluation metrics such as Meteor (Banerjee and Lavie, 2005), and to incorporate features that can measure syntactic similarities in training the classifier, in the spirit of (Owczarzak et al., 2007). Currently, only a standard phrase-based SMT system is used, so we plan to test our method on a hierarchical system (Chiang, 2005) to facilitate direct comparison with (Koehn and Senellart, 2010). We will also carry out experiments on other data sets and for more language pairs.

Acknowledgments

This work is supported by Science Foundation Ireland (Grant No 07/CE/I1142) and part funded under FP7 of the EC within the EuroMatrix+ project (grant No 231720). The authors would like to thank the reviewers for their insightful comments and suggestions.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, MI.

Ergun Biçici and Marc Dymetman. 2008. Dynamic translation memory: Using statistical machine translation to improve translation memory. In *Proceedings of the 9th Internation Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 454–465, Haifa, Israel.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

David Chiang. 2005. A hierarchical Phrase-Based model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, MI.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Yifan He, Yanjun Ma, Josef van Genabith, and Andy Way. 2010. Bridging SMT and TM with translation recommendation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 622–630, Uppsala, Sweden.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 181–184, Detroit, MI.

Philipp Koehn and Jean Senellart. 2010. Convergence of translation memory and statistical machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31, Denver, CO.

Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics*, pages 48–54, Edmonton, AB, Canada.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Vol-*

ume Proceedings of the Demo and Poster Sessions, pages 177–180, Prague, Czech Republic.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. 2007. A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276.

Adam Lopez. 2008. Tera-scale translation models via pattern matching. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 505–512, Manchester, UK, August.

Eric W. Noreen. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. Wiley-Interscience, New York, NY.

Franz Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan.

Karolina Owczarzak, Josef van Genabith, and Andy Way. 2007. Labelled dependencies in machine translation evaluation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 104–111, Prague, Czech Republic.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of Machine Translation. In *40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA.

John C. Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, pages 61–74.

Richard Sikes. 2007. Fuzzy matching in theory and practice. *Multilingual*, 18(6):39–43.

Michel Simard and Pierre Isabelle. 2009. Phrase-based machine translation in a computer-assisted translation environment. In *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*, pages 120 – 127, Ottawa, Ontario, Canada.

James Smith and Stephen Clark. 2009. EBMT for SMT: A new EBMT-SMT hybrid. In *Proceedings of the 3rd International Workshop on Example-Based Machine Translation*, pages 3–10, Dublin, Ireland.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas (AMTA-2006)*, pages 223–231, Cambridge, MA, USA.

Lucia Specia, Craig Saunders, Marco Turchi, Zhuoran Wang, and John Shawe-Taylor. 2009. Improving the confidence of machine translation quality estimates. In *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*, pages 136 – 143, Ottawa, Ontario, Canada.

Andreas Stolcke. 2002. SRILM – An extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904, Denver, CO.

Ventsislav Zhechev and Josef van Genabith. 2010. Seeding statistical machine translation with translation memory output through tree-based structural alignment. In *Proceedings of the Fourth Workshop on Syntax and Structure in Statistical Translation*, pages 43–51, Beijing, China.