Instance Selection for Machine Translation using Feature Decay Algorithms

Ergun Biçici
Koç University
34450 Sariyer, Istanbul, Turkey
ebicici@ku.edu.tr

Deniz Yuret
Koç University
34450 Sariyer, Istanbul, Turkey
dyuret@ku.edu.tr

Abstract

We present an empirical study of instance selection techniques for machine translation. In an active learning setting, instance selection minimizes the human effort by identifying the most informative sentences for translation. In a transductive learning setting, selection of training instances relevant to the test set improves the final translation quality. After reviewing the state of the art in the field, we generalize the main ideas in a class of instance selection algorithms that use feature decay. Feature decay algorithms increase diversity of the training set by devaluing features that are already included. We show that the feature decay rate has a very strong effect on the final translation quality whereas the initial feature values, inclusion of higher order features, or sentence length normalizations do not. We evaluate the best instance selection methods using a standard Moses baseline using the whole 1.6 million sentence English-German section of the Europarl corpus. We show that selecting the best 3000 training sentences for a specific test sentence is sufficient to obtain a score within 1 BLEU of the baseline, using 5% of the training data is sufficient to exceed the baseline, and a ~ 2 BLEU improvement over the baseline is possible by optimally selected subset of the training data. In out-of-domain translation, we are able to reduce the training set size to about 7% and achieve a similar performance with the baseline.

1 Introduction

Statistical machine translation (SMT) makes use of a large number of parallel sentences, sentences whose translations are known in the target language, to derive translation tables, estimate parameters, and generate the actual translation. Not all of the parallel corpus nor the translation table that is generated is used during decoding a given set of test sentences and filtering is usually performed for computational advantage (Koehn et al., 2007). Some recent regression-based statistical machine translation systems rely on a small sized training data to learn the mappings between source and target features (Wang and Shawe-Taylor, 2008; Serrano et al., 2009; Bicici and Yuret, 2010). Regression has some computational disadvantages when scaling to large number of training instances.

Previous work shows that the more the training data, the better the translations become (Koehn, 2006). However, with the increased size of the parallel corpus there is also the added noise, making relevant instance selection important. Phrase-based SMT systems rely heavily on accurately learning word alignments from the given parallel corpus. Proper instance selection plays an important role in obtaining a small sized training set with which correct alignments can be learned. Word-level translation accuracy is also affected by the number of times a word occurs in the parallel corpus (Koehn and Knight, 2001). Koehn and Knight find that about 50 examples per word are required to achieve a performance close to using a bilingual lexicon in their experiments. Translation performance can improve as we include multiple possible translations for a given word, which increases

Proceedings of the 6th Workshop on Statistical Machine Translation, pages 272–283,
Edinburgh, Scotland, UK, July 30–31, 2011.

the diversity of the training set.

Transduction uses test instances, which can sometimes be accessible at training time, to learn specific models tailored towards the test set which also reduces computation by not using the full training set. Transductive retrieval selects training data close to the test set given a parallel corpus and a test set. This work shows that *transductive retrieval* of the training set for statistical machine translation allows us to achieve a performance better than using all of the parallel corpus. When selecting training data, we seek to maximize the coverage or the percentage of test source and target features (i.e. n-grams) found in the training set using minimal number of target training features and a fixed number of training instances. Diversifying the set of training sentences can help us increase the coverage. We show that target coverage bounds the achievable BLEU score with a given training set and small increases can result in large increases on this BLEU bound.

We develop the feature decay algorithms (FDA) that aim to maximize the coverage of the target language features and achieve significant gains in translation performance. We find that decaying feature weights has significant effect on the performance. We achieve improvements of $\sim$2 BLEU points using about 20% of the available training data in terms of target words and $\sim$1 BLEU points with only about 5%. We show that selecting 3000 instances for a test sentence is sufficient to obtain a score within 1 BLEU of the baseline. In the out-of-domain translation task, we are able to reduce the training set size to its 7% to achieve a similar performance with the baseline.

The next section reviews related previous work. We discuss the FDA in section 3. Section 4 presents our coverage and translation results both in and out-of-domain and includes an instance selection method also designed for improving word alignment results. We list our contributions in the last section.

2 Related Work

Transductive learning makes use of test instances, which can sometimes be accessible at training time, to learn specific models tailored towards the test set. Selection of training instances relevant to the test set improves the final translation quality as in transductive learning and decreases human effort by identifying the most informative sentences for translation as in active learning. Instance selection in a transductive learning framework selects the best instances for a given test set (Lü et al., 2007). *Active learning* selects training samples that will benefit the learning algorithm the most over the unlabeled dataset $\mathcal{U}$ from a labeled training set $\mathcal{L}$ or from $\mathcal{U}$ itself after labeling (Banko and Brill, 2001). Active learning in SMT selects which instances to add to the training set to improve the performance of a baseline system (Haffari et al., 2009; Ananthakrishnan et al., 2010). Recent work involves selecting sentence or phrase translation tasks for external human effort (Bloodgood and Callison-Burch, 2010). Below we present examples of both with a label indicating whether they follow an approach close to active learning [AL] or transductive learning [TL] and in our experiments we use the transductive framework.

TF-IDF [TL]: Lü et al. (2007) use tf-idf information retrieval technique based cosine score to select a subset of the parallel corpus close to the test set for SMT training. They outperform the baseline system when the top 500 training instances per test sentence are selected. The terms used in their tf-idf measure correspond to words where this work focuses on bigram feature coverage. When the combination of the top N selected sentences are used as the training set, they show increase in the performance at the beginning and decrease when 2000 sentences are selected for each test sentence.

N-gram coverage [AL]: Eck et al. (2005) use n-gram feature coverage to sort and select training instances using the following score:

$$\phi_{NGRAM}(S) = \frac{\sum_{i=1}^{n} \sum_{\text{unseen } x \in X_i(S)} C(x)}{|S|}, \quad (1)$$

for sentence S with $X_i(S)$ storing the i-grams found in S and $C(x)$ returning the count of x in the parallel corpus. ϕ_{NGRAM} score sums over unseen n-grams to increase the coverage of the training set. The denominator involving the length of the sentence takes the translation cost of the sentence into account. Eck et al. (2005) also note that longer sentences are more difficult for training SMT models. In their experiments, they are not able to reach a performance above the baseline

system's BLEU score, which is using all of the parallel corpus, but they achieve close performance by using about 15% of the parallel corpus.

DWDS [AL]: Density weighted diversity sampling (*DWDS*) (Ambati et al., 2010) score tries to select sentences containing the n-gram features in the unlabeled dataset $\mathcal{U}$ while increasing the diversity among the sentences selected, $\mathcal{L}$ (labeled). *DWDS* increases the score of a sentence with increasing frequency of its n-grams found in $\mathcal{U}$ and decreases with increasing frequency in the already selected set of sentences, $\mathcal{L}$, in favor of diversity. Let $P_{\mathcal{U}}(x)$ denote the probability of feature x in $\mathcal{U}$ and $C_{\mathcal{L}}(x)$ denote its count in $\mathcal{L}$. Then:

$$d(S) = \frac{\sum_{x \in X(S)} P_{\mathcal{U}}(x) e^{-\lambda C_{\mathcal{L}}(x)}}{|X(S)|} \quad (2)$$

$$u(S) = \frac{\sum_{x \in X(S)} I(x \notin X(\mathcal{L}))}{|X(S)|} \quad (3)$$

$$\phi_{DWDS}(S) = \frac{2d(S)u(S)}{d(S) + u(S)}, \quad (4)$$

where $X(S)$ stores the features of S and λ is a decay parameter. $d(S)$ denotes the density of S proportional to the probability of its features in $\mathcal{U}$ and inversely proportional to their counts in $\mathcal{L}$ and $u(S)$ its uncertainty, measuring the percentage of new features in S. These two scores are combined using harmonic mean. *DWDS* tries to select sentences containing similar features in $\mathcal{U}$ with high diversity. In their active learning experiments, they selected 1000 training instances in each iteration and retrained the SMT system.

Log-probability ratios [AL]: Haffari et al. (2009) develop sentence selection scores using feature counts in $\mathcal{L}$ and $\mathcal{U}$, increasing for frequent features in $\mathcal{U}$ and decreasing for frequent features in $\mathcal{L}$. They use geometric and arithmetic averages of log-probability ratios in an active learning setting where 200 sentences from $\mathcal{U}$ are selected and added to $\mathcal{L}$ with their translations for 25 iterations (Haffari et al., 2009). Later, Haffari et al. (2009) distinguish between features found in the phrase table, x_{reg}, and features not found, x_{oov}. OOV features are segmented into subfeatures (i.e. feature "go to school" is segmented as: (go to school), (go)(to school), (go to)(school), (go)(to)(school)). *Expected log probability ratio (ELPR)* score is used:

$$\phi_{ELPR}(S) = \frac{0.4}{|X_{reg}(S)|} \sum_{x \in X_{reg}(S)} \log \frac{P_{\mathcal{U}}(x)}{P_{\mathcal{L}}(x)} + \frac{0.6}{|X_{oov}(S)|} \sum_{x \in X_{oov}(S)} \sum_{h \in H(x)} \frac{1}{|H(x)|} \sum_{y \in Y_h(x)} \log \frac{P_{\mathcal{U}}(y)}{P_{\mathcal{L}}(y)}, \quad (5)$$

where $H(x)$ return the segmentations of x and $Y_h(x)$ return the features found in segment h. ϕ_{ELPR} performs better than geometric average in their experiments (Haffari and Sarkar, 2009).

Perplexity [AL & TL]: Perplexity of the training instance as well as inter-SMT-system disagreement are also used to select training data for translation models (Mandal et al., 2008). The increased difficulty in translating a parallel sentence or its novelty as found by the perplexity adds to its importance for improving the SMT model's performance. A sentence having high perplexity (a rare sentence) in $\mathcal{L}$ and low perplexity (a common sentence) in $\mathcal{U}$ is considered as a candidate for addition. They are able to improve the performance of a baseline system trained on some initial corpus together with additional parallel corpora using the initial corpus and part of the additional data.

Alignment [TL]: Uszkoreit et al. (2010) mine parallel text to improve the performance of a baseline translation model on some initial document translation tasks. They retrieve similar documents using inverse document frequency weighted cosine similarity. Then, they filter nonparallel sentences using their word alignment performance, which is estimated using the following score:

$$\text{score}(A) = \sum_{(s,t) \in A} \ln \frac{p(s,t)}{p(s)p(t)}, \quad (6)$$

where A stands for an alignment between source and target words and the probabilities are estimated using a word aligned corpus. The produced parallel data is used to expand a baseline parallel corpus and shown to improve the translation performance of machine translation systems.

3 Instance Selection with Feature Decay

In this section we will describe a class of instance selection algorithms for machine translation that

use feature decay, i.e. increase the diversity of the training set by devaluing features that have already been included. Our abstraction makes three components of such algorithms explicit permitting experimentation with their alternatives:

- The value of a candidate training sentence as a function of its features.
- The initial value of a feature.
- The update of the feature value as instances are added to the training set.

A feature decay algorithm (FDA) aims to maximize the coverage of the target language features (such as words, bigrams, and phrases) for the test set. A target language feature that does not appear in the selected training instances will be difficult to produce regardless of the decoding algorithm (impossible for unigram features). In general we do not know the target language features, only the source language side of the test set is available. Unfortunately, selecting a training instance with a particular source language feature does not guarantee the coverage of the desired target language feature. There may be multiple translations of a feature appropriate for different senses or different contexts. For each source language feature in the test set, FDA tries to find as many training instances as possible to increase the chances of covering the appropriate target language feature. It does this by reducing the value of the features that are already included after picking each training instance. Algorithm 1 gives the pseudo-code for FDA.

Algorithm 1: The Feature Decay Algorithm

Input: Bilingual corpus $\mathcal{U}$, test set features $\mathcal{F}$, and desired number of training instances N.
Data: A priority queue $\mathcal{Q}$, sentence scores `score`, feature values `fvalue`.
Output: Subset of the corpus to be used as the training data $\mathcal{L} \subseteq \mathcal{U}$.

```
1  foreach f ∈ F do
2    fvalue(f) ← init(f,U)
3  foreach S ∈ U do
4    score(S) ← Σ_{f∈features(S)} fvalue(f)
5    push(Q, S, score(S))
6  while |L| < N do
7    S ← pop(Q)
8    score(S) ← Σ_{f∈features(S)} fvalue(f)
9    if score(S) ≥ topval(Q) then
10     L ← L ∪ {S}
11     foreach f ∈ features(S) do
12       fvalue(f) ← decay(f,U,L)
13   else
14     push(Q, S, score(S))
```

The input to the algorithm is a parallel corpus, the number of desired training instances, and the source language features of the test set. We use unigram and bigram features; adding trigram features does not seem to significantly affect the results. The user has the option of running the algorithm for each test sentence separately, then possibly combining the resulting training sets. We will present results with these variations in Section 4.

The first foreach loop initializes the value of each test set feature. We experimented with initial feature values that are constant, proportional to the length of the n-gram, or log-inverse of the corpus frequency. We have observed that the initial value does not have a significant effect on the quality of training instances selected. The feature decay rule dominates the behavior of the algorithm after the first few iterations. However, we prefer the log-inverse values because they lead to fewer score ties among candidate instances and result in faster running times.

The second foreach loop initializes the score for each candidate training sentence and pushes them onto a priority queue. The score is calculated as the sum of the feature values. Note that as we change the feature values, the sentence scores in the priority queue will no longer be correct. However they will still be valid upper bounds because the feature values only get smaller. Features that do not appear in the test set are considered to have zero value. This observation can be used to speed up the initialization by using a feature index and only iterating over the sentences that have features in common with the test set.

Finally the while loop populates the training set by picking candidate sentences with the highest scores. This is done by popping the top scoring candidate S from the priority queue at each iteration. We recalculate its score because the values

of its features may have changed. We compare the recalculated score of S with the score of the next best candidate. If the score of S is equal or better we are sure that it is the top candidate because the scores in the priority queue are upper bounds. In this case we place S in our training set and decay the values of its features. Otherwise we push S back on the priority queue with its updated score.

The feature decay function on Line 12 is the heart of the algorithm. Unlike the choice of features (bigram vs trigram) or their initial values (constant vs log–inverse–frequency) the rate of decay has a significant effect on the performance. We found it is optimal to reduce feature values at a rate of $1/n$ where n is the current training set count of the feature. The results get significantly worse with no feature decay. They also get worse with faster, exponential feature decay, e.g. $1/2^n$. Table 1 presents the experimental results that support these conclusions. We use the following settings for the experiments in Section 4:

$$\texttt{init}(f,\mathcal{U}) = 1 \text{ or } \log(|\mathcal{U}|/\texttt{cnt}(f,\mathcal{U}))$$

$$\texttt{decay}(f,\mathcal{U},\mathcal{L}) = \frac{\texttt{init}(f,\mathcal{U})}{1+\texttt{cnt}(f,\mathcal{L})} \text{ or } \frac{\texttt{init}(f,\mathcal{U})}{1+2^{\texttt{cnt}(f,\mathcal{L})}}$$

init	decay	en→de		de→en	
1	none	.761	.484	.698	.556
$\log(1/f)$	none	.855	.516	.801	.604
1	$1/n$	**.967**	**.575**	**.928**	**.664**
$\log(1/f)$	$1/n$	.967	.570	.928	.656
1	$1/2^n$	.967	.553	.928	.653
$\log(1/f)$	$1/2^n$	.967	.557	.928	.651

Table 1: FDA experiments. The first two columns give the initial value and decay formula used for features. f is the corpus frequency of a feature and n is its count in selected instances. The next four columns give the expected coverage of the source and target language bigrams of a test sentence when 100 training sentences are selected.

4 Experiments

We perform translation experiments on the English-German language pair using the parallel corpus provided in WMT'10 (Callison-Burch et al., 2010). The English-German section of the Europarl corpus contains about 1.6 million sentences. We perform *in-domain* experiments to discriminate among different instance selection techniques better in a setting with low out-of-vocabulary rate. We randomly select the test set *test* with $2,588$ target words and separate development set *dev* with $26,178$ target words. We use the language model corpus provided in WMT'10 (Callison-Burch et al., 2010) to build a 5-gram model.

We use target language *bigram* coverage, $tcov$, as a quality measure for a given training set, which measures the percentage of the target bigram features of the test sentence found in a given training set. We compare $tcov$ and the translation performance of FDA with related work. We also perform small scale SMT experiments where only a couple of thousand training instances are used for each test sentence.

4.1 The Effect of Coverage on Translation

BLEU (Papineni et al., 2001) is a precision based measure and uses n-gram match counts up to order n to determine the quality of a given translation. The absence of a given word or translating it as another word interrupts the continuity of the translation and decreases the BLEU score even if the order among the words is determined correctly. Therefore, the target coverage of an out-of-domain test set whose translation features are not found in the training set bounds the translation performance of an SMT system.

We estimate this translation performance bound from target coverage by assuming that the missing tokens can appear randomly at any location of a given sentence where sentence lengths are normally distributed with mean 25.6 and standard deviation 14.1. This is close to the sentence length statistics of the German side Europarl corpus used in WMT'10 (WMT, 2010). We replace all unknown words found with an UNK token and calculate the BLEU score. We perform this experiment for $10,000$ instances and repeat for 10 times.

The obtained BLEU scores for target coverage values is plotted in Figure 1 with label *estimate*. We also fit a third order polynomial function of target coverage 0.025 BLEU scores above the *estimate* values to show the similarity with the

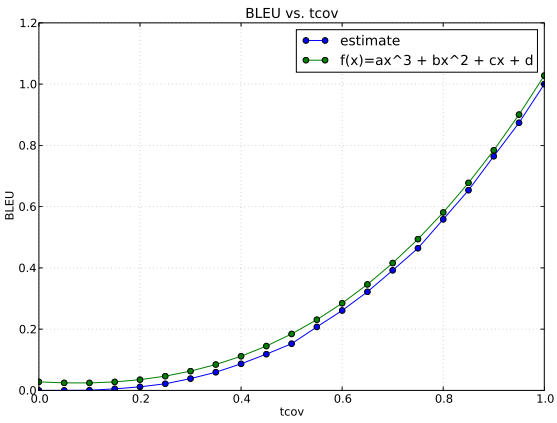

Figure 1: Effect of coverage on translation performance. BLEU bound is a third-order function of target coverage. High coverage → High BLEU.

BLEU scores bound estimated, whose parameters are found to be $[0.56, 0.53, -0.09, 0.003]$ with a least-squares fit. Figure 1 shows that the BLEU score bound obtained has a third-order polynomial relationship with target coverage and small increases in the target coverage can result in large increases on this BLEU bound.

4.2 Coverage Results

We select N training instances per test sentence using FDA (Algorithm 1), *TF-IDF* with bigram features, *NGRAM* scoring (Equation 1), *DWDS* (Equation 4), and *ELPR* (Equation 5) techniques from previous work. For the active learning algorithms, source side test corpus becomes $\mathcal{U}$ and the selected training set $\mathcal{L}$. For all the techniques, we compute 1-grams and 2-grams as the features used in calculating the scores and add only one sentence to the training set at each iteration except for *TF-IDF*. We set λ parameter of *DWDS* to 1 as given in their paper. We adaptively select the top scoring instance at each step from the set of possible sentences $\mathcal{U}$ with a given scorer $\phi(.)$ and add the instance to the training set, $\mathcal{L}$, until the size of $\mathcal{L}$ reaches N for the related work other than *TF-IDF*. We test all algorithms in this transductive setting.

We measure the *bigram* coverage when all of the training sentences selected for each test sentence are combined. The results are presented in Figure 2 where the x-axis is the number of words of the training set and y-axis is the target coverage obtained. FDA has a steep slope in its increase and it is able to reach target coverage of ~ 0.84. *DWDS* performs worse initially but its target coverage improve after a number of instances are selected due to its exponential feature decay procedure. *TF-IDF* performs worse than *DWDS* and it provides a fast alternative to FDA instance selection but with some decrease in coverage. *ELPR* and *NGRAM* instance selection techniques perform worse. *NGRAM* achieves better coverage than *ELPR*, although it lacks a decay procedure.

When we compare the sentences selected, we observe that FDA prefers longer sentences due to summing feature weights and it achieves larger target coverage value. *NGRAM* is not able to discriminate between sentences well and a lot of sentences of the same length get the same score when the unseen n-grams belong to the same frequency class. The statistics of $\mathcal{L}$ obtained with the instance selection techniques differ from each other as given in Table 2, where $N = 1000$ training instances selected per test sentence. We observe that *DWDS* has fewer unique target bigram features than *TF-IDF* although it selects longer target sentences. *NGRAM* obtains a large number of unique target bigrams although its selected target sentences have similar lengths with *DWDS* and *ELPR* prefers short sentences.

Technique	Unique bigrams	Words per sent	$tcov$
FDA	827,928	35.8	.74
DWDS	412,719	16.7	.67
TF-IDF	475,247	16.2	.65
NGRAM	626,136	16.6	.55
ELPR	172,703	10.9	.35

Table 2: Statistics of the obtained target $\mathcal{L}$ for $N = 1000$.

4.3 Translation Results

We develop separate phrase-based SMT models using Moses (Koehn et al., 2007) using default settings with maximum sentence length set to 80 and obtained baseline system score as 0.3577 BLEU. We use the training instances selected by FDA in

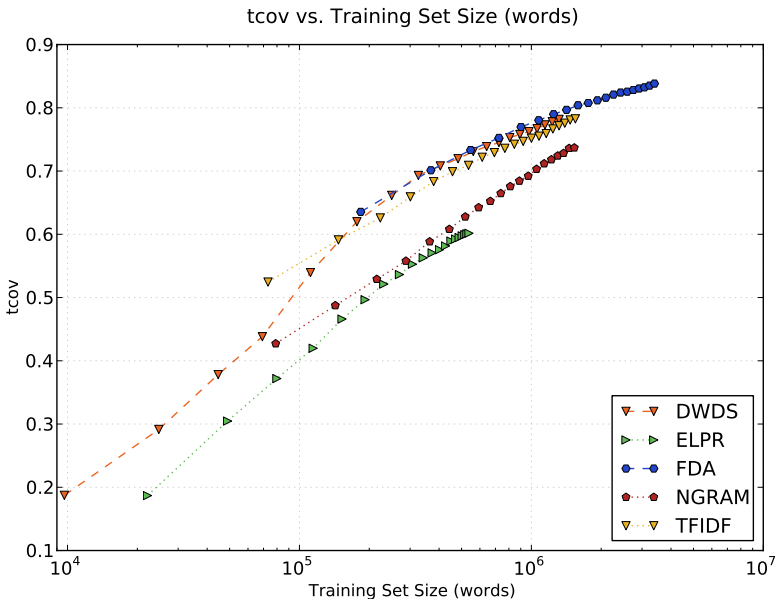

Figure 2: Target coverage curve comparison with previous work. Figure shows the rate of increase in $tcov$ as the size of $\mathcal{L}$ increase.

three learning settings:

- $\mathcal{L}_{\cup}$ $\mathcal{L}$ is the union of the instances selected for each test sentence.
- $\mathcal{L}_{\cup_{\mathcal{F}}}$ $\mathcal{L}$ is selected using all of the features found in the test set.
- $\mathcal{L}_{\mathcal{I}}$ $\mathcal{L}$ is the set of instances selected for each test sentence.

We develop separate Moses systems with each training set and $\mathcal{L}_{\mathcal{I}}$ corresponds to developing a Moses system for each test sentence. $\mathcal{L}_{\cup}$ results are plot in Figure 3 where we increasingly select $N \in \{100, 200, 500, 1000, 2000, 3000, 5000, 10000\}$ instances for each test sentence for training. The improvements over the baseline are statistically significant with paired bootstrap resampling using 1000 samples (Koehn, 2004). As we select more instances, the performance of the SMT system increases as expected and we start to see a decrease in the performance after selecting $\sim 10^7$ target words. We obtain comparable results for the *de-en* direction. The performance increase is likely to be due to the reduction in the number of noisy or irrelevant training instances and the increased precision in the probability estimates in the generated phrase tables.

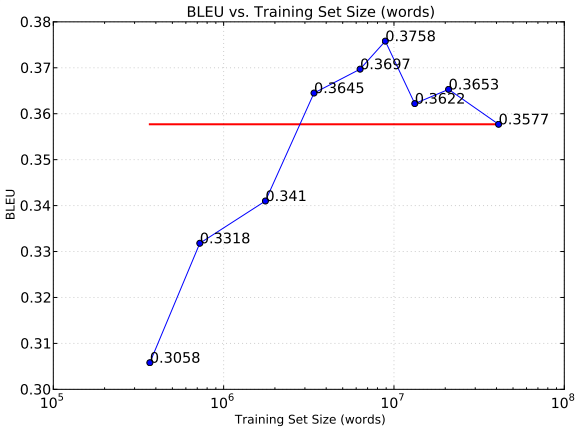

Figure 3: BLEU vs. the number of target words in $\mathcal{L}_{\cup}$.

$\mathcal{L}_{\cup_{\mathcal{F}}}$ results given in Table 3 show that we can achieve within 1 BLEU performance using about 3% of the parallel corpus target words (30,000 instances) and better performance using only about 5% (50,000 instances).

The results with $\mathcal{L}_{\mathcal{I}}$ when building an individ-

# sent	# target words	BLEU	NIST
10,000	449,116	0.3197	5.7788
20,000	869,908	0.3417	6.0053
30,000	1,285,096	0.3492	6.0246
50,000	**2,089,403**	**0.3711**	**6.1561**
100,000	**4,016,124**	**0.3648**	**6.1331**
ALL	41,135,754	0.3577	6.0653

Table 3: Performance for *en-de* using $\mathcal{L}_{\cup_{\mathcal{F}}}$. ALL corresponds to the baseline system using all of the parallel corpus. **bold** correspond to statistically significant improvement over the baseline result.

ual Moses model for each test sentence are given in Table 4. Individual SMT training and translation can be preferable due to smaller computational costs and high parallelizability. As we translate a single sentence with each SMT system, tuning weights becomes important. We experiment three settings: (1) using 100 sentences for tuning, which are randomly selected from *dev.1000*, (2) using the mean of the weights obtained in (1), and (3) using the weights obtained in the union learning setting ($\mathcal{L}_{\cup}$). We observe that we can obtain a performance within 2 BLEU difference to the baseline system by training on 3000 instances per sentence (underlined) using the mean weights and 1 BLEU difference using the union weights. We also experimented with increasing the N-best list size used during MERT optimization (Hasan et al., 2007), with increased computational cost, and observed some increase in the performance.

N	100 dev sents	Mean	Union
1000	0.3149	0.3242	0.3354
2000	0.3258	0.3352	0.3395
3000	0.3270	0.3374	0.3501
5000	0.3217	0.3303	0.3458

Table 4: $\mathcal{L}_{\mathcal{I}}$ performance for *en-de* using 100 sentences for tuning or mean of the weights or dev weights obtained with the union setting.

Comparison with related work: Table 5 presents the translation results compared with previous work selecting 1000 instances per test sentence. We observe that coverage and translation performance are correlated. Although the coverage increase of *DWDS* and FDA appear similar, due to the third-order polynomial growth of BLEU with respect to coverage, we achieve large BLEU gains in translation. We observe increased BLEU gains when compared with the results of *TF-IDF*, *NGRAM*, and *ELPR* in order.

FDA	*DWDS*	*TF-IDF*	*NGRAM*	*ELPR*
0.3645	0.3547	0.3405	0.2572	0.2268

Table 5: BLEU results using different techniques with $N = 1000$. High coverage $\rightarrow$ High BLEU.

We note that *DWDS* originally selects instances using the whole test corpus to estimate $P_{\mathcal{U}}(x)$ and selects 1000 instances at each iteration. We experimented with both of these settings and obtained 0.3058 and 0.3029 BLEU respectively. Lower performance suggest the importance of updating weights after each instance selection step.

4.4 Instance Selection for Alignment

We have shown that high coverage is an integral part of training sets for achieving high BLEU performance. SMT systems also heavily rely on the word alignment of the parallel corpus to derive a phrase table that can be used for translation. GIZA++ (Och and Ney, 2003) is commonly used for word alignment and phrase table generation, which is prone to making more errors as the length of the training sentence increase (Ravi and Knight, 2010). Therefore, we analyze instance selection techniques that optimize coverage and word alignment performance and at the same time do not produce very long sentences. Too few words per sentence may miss the phrasal structure, whereas too many words per sentence may miss the actual word alignment for the features we are interested. We are also trying to retrieve relevant training sentences for a given test sentence to increase the feature alignment performance.

Shortest: A baseline strategy that can minimize the training feature set's size involves selecting the shortest translations containing each feature.

Co-occurrence: We use *co-occurrence* of words in the parallel corpus to retrieve sentences containing co-occurring items. Dice's coefficient (Dice, 1945) is used as a heuristic word alignment technique giving an association score for each pair of word positions (Och and Ney, 2003).

We define Dice's coefficient score as:

$$dice(x, y) = \frac{2C(x, y)}{C(x)C(y)}, \quad (7)$$

where $C(x, y)$ is the number of times x and y co-occur and $C(x)$ is the count of observing x in the selected training set. Given a test source sentence, $S_{\mathcal{U}}$, we can estimate the goodness of a training sentence pair, (S, T), by the sum of the alignment scores:

$$\phi_{dice}(S_{\mathcal{U}}, S, T) = \frac{\sum_{x \in X(S_{\mathcal{U}})} \sum_{j=1}^{|T|} \sum_{y \in Y(x)} dice(y, T_j)}{|T| \log |S|}, \quad (8)$$

where $X(S_{\mathcal{U}})$ stores the features of $S_{\mathcal{U}}$ and $Y(x)$ lists the tokens in feature x. The difficulty of word aligning a pair of training sentences, (S, T), can be approximated by $|S|^{|T|}$. We use a normalization factor proportional to $|T| \log |S|$.

The average target words per sentence using ϕ_{dice} drops to 26.2 compared to 36.3 of FDA. We still obtain a better performance than the baseline *en-de* system with the union of 1000 training instances per sentence with 0.3635 BLEU and 6.1676 NIST scores. Coverage comparison with FDA shows slight improvement with lower number of target bigrams and similar trend for others (Figure 4). We note that shortest strategy achieves better performance than both *ELPR* and *NGRAM*. We obtain 0.3144 BLEU and 5.5 NIST scores in the individual translation task with 1000 training instances per sentence and 0.3171 BLEU and 5.4662 NIST scores when the mean of the weights is used.

4.5 Out-of-domain Translation Results

We have used FDA and *dice* algorithms to select training sets for the out-of-domain challenge test sets used in (Callison-Burch et al., 2011). The parallel corpus contains about 1.9 million training sentences and the test set contain 3003 sentences. We built separate Moses systems using all of the parallel corpus for the language pairs *en-de*, *de-en*, *en-es*, and *es-en*. We created training sets using all of the features of the test set to select training instances. The results given in Table 6 show that we can achieve similar BLEU performance using about 7% of the parallel corpus target words (200,000 instances) using *dice* and about 16% using FDA. In the out-of-domain translation task, we are able to reduce the training set size to achieve a performance close to the baseline. The sample points presented in the table is chosen proportional to the relative sizes of the parallel corpus sizes of WMT'10 and WMT'11 datasets and the training set size of the peak in Figure 3. We may be able to achieve better performance in the out-of-domain task as well. The sample points in Table 6 may be on either side of the peak.

5 Contributions

We have introduced the feature decay algorithms (FDA), a class of instance selection algorithms that use feature decay, which achieves better target coverage than previous work and achieves significant gains in translation performance. We find that decaying feature weights has significant effect on the performance. We demonstrate that target coverage and translation performance are correlated, showing that target coverage is also a good indicator of BLEU performance. We have shown that target coverage provides an upper bound on the translation performance with a given training set.

We achieve improvements of $\sim$2 BLEU points using about 20% of the available training data in terms of target words with FDA and $\sim$ 1 BLEU points with only about 5%. We have also shown that by training on only 3000 instances per sentence we can reach within 1 BLEU difference to the baseline system. In the out-of-domain translation task, we are able to reduce the training set size to achieve a similar performance with the baseline.

Our results demonstrate that SMT systems can improve their performance by transductive training set selection. We have shown how to select instances and achieved significant performance improvements.

References

Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. 2010. Active learning and crowd-sourcing for machine translation. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Language Resources Association (ELRA).

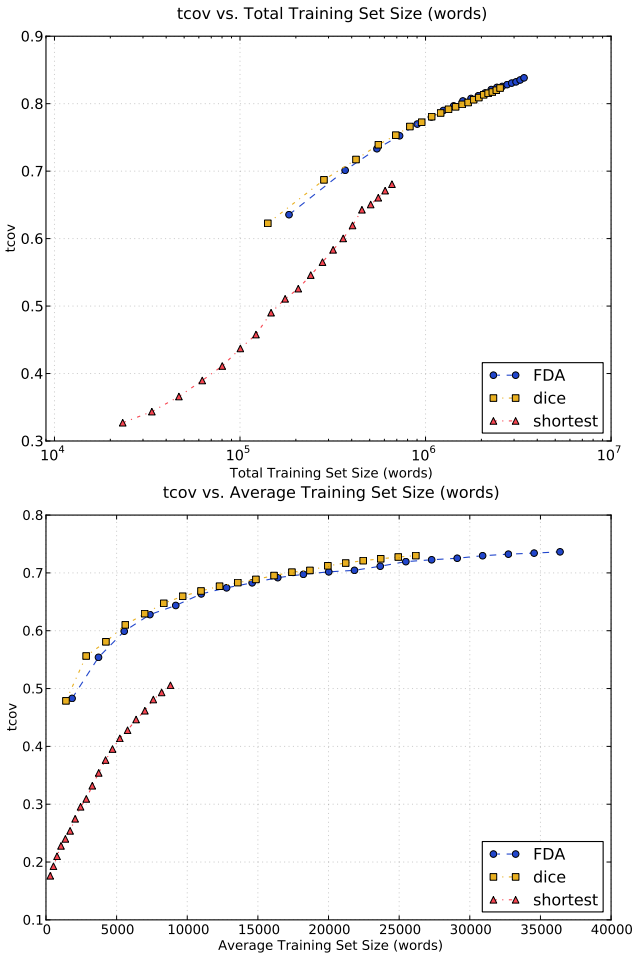

Figure 4: Target coverage per target words comparison. Figure shows the rate of increase in $tcov$ as the size of $\mathcal{L}$ increase. Target coverage curves for total training set size is given on the left plot and for average training set size per test sentence on the right plot.

		en-de	*de-en*	*en-es*	*es-en*
BLEU	ALL	0.1376	0.2074	0.2829	0.2919
	FDA	0.1363	0.2055	0.2824	0.2892
	dice	0.1374	0.2061	0.2834	0.2857
# target words $\times 10^6$	ALL	47.4	49.6	52.8	50.4
	FDA	7.9	8.0	8.7	8.2
	dice	6.9	7.0	3.9	3.6
% of ALL	FDA	17	16	16	16
	dice	14	14	7.4	7.1

Table 6: Performance for the out-of-domain task of (Callison-Burch et al., 2011). ALL corresponds to the baseline system using all of the parallel corpus.

Sankaranarayanan Ananthakrishnan, Rohit Prasad, David Stallard, and Prem Natarajan. 2010. Discriminative sample selection for statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 626–635, Cambridge, MA, October. Association for Computational Linguistics.

Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse, France, July. Association for Computational Linguistics.

Ergun Bicici and Deniz Yuret. 2010. L_1 regularized regression for reranking and system combination in machine translation. In *Proceedings of the ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, Uppsala, Sweden, July. Association for Computational Linguistics.

Michael Bloodgood and Chris Callison-Burch. 2010. Bucking the trend: Large-scale cost-focused active learning for statistical machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 854–864, Uppsala, Sweden, July. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, and Omar Zaidan, editors. 2010. *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Association for Computational Linguistics, Uppsala, Sweden, July.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, and Omar Zaidan, editors. 2011. *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, England, July.

Lee R. Dice. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. Low cost portability for statistical machine translation based on n-gram coverage. In *Proceedings of the 10th Machine Translation Summit, MT Summit X*, pages 227–234, Phuket, Thailand, September.

Gholamreza Haffari and Anoop Sarkar. 2009. Active learning for multilingual statistical machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 181–189, Suntec, Singapore, August. Association for Computational Linguistics.

Gholamreza Haffari, Maxim Roy, and Anoop Sarkar. 2009. Active learning for statistical phrase-based machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 415–423, Boulder, Colorado, June. Association for Computational Linguistics.

Saša Hasan, Richard Zens, and Hermann Ney. 2007. Are very large N-best lists useful for SMT? In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 57–60, Rochester, New York, April. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2001. Knowledge sources for word-level translation models. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*.

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 *Annual Meeting of the Assoc. for Computational Linguistics*, pages 177–180, Prague, Czech Republic, June.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Philipp Koehn. 2006. Statistical machine translation: the basic, the novel, and the speculative. Tutorial at EACL 2006.

Yajuan Lü, Jin Huang, and Qun Liu. 2007. Improving statistical machine translation performance by training data selection and optimization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 343–350, Prague, Czech Republic, June. Association for Computational Linguistics.

A. Mandal, D. Vergyri, W. Wang, J. Zheng, A. Stolcke, G. Tur, D. Hakkani-Tur, and N.F. Ayan. 2008. Efficient data selection for machine translation. In *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE*, pages 261 –264.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for*

Computational Linguistics, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Sujith Ravi and Kevin Knight. 2010. Does giza++ make search errors? *Computational Linguistics*, 36(3):295–302.

Nicolas Serrano, Jesus Andres-Ferrer, and Francisco Casacuberta. 2009. On a kernel regression approach to machine translation. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 394–401.

Jakob Uszkoreit, Jay Ponte, Ashok Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1101–1109, Beijing, China, August. Coling 2010 Organizing Committee.

Zhuoran Wang and John Shawe-Taylor. 2008. Kernel regression framework for machine translation: UCL system description for WMT 2008 shared translation task. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 155–158, Columbus, Ohio, June. Association for Computational Linguistics.

WMT. 2010. ACL Workshop: Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR, July.