Optimizing Instance Selection for Statistical Machine Translation with Feature Decay Algorithms

***Abstract*—We introduce FDA5 for efficient parameterization, optimization, and implementation of feature decay algorithms (FDA), a class of instance selection algorithms that use feature decay. FDA increase the diversity of the selected training set by devaluing features (i.e. n-grams) that have already been included. FDA5 decides which instances to select based on three functions used for initializing and decaying feature values and scaling sentence scores controlled with 5 parameters. We present optimization techniques that allow FDA5 to adapt these functions to in-domain and out-of-domain translation tasks for different language pairs. In a transductive learning setting, selection of training instances relevant to the test set can improve the final translation quality. In machine translation experiments performed on the 2 million sentence English-German section of the Europarl corpus, we show that a subset of the training set selected by FDA5 can gain up to 3.22 BLEU points compared to a randomly selected subset of the same size, can gain up to 0.43 BLEU points compared to using all of the available training data, and can reach within 0.5 BLEU to the full training set result by using only 2.7% of the full training data. In an active learning setting, FDA5 minimizes the human effort by identifying the most informative sentences for translation and FDA gains up to 0.45 BLEU points compared to using all of the available training data and 1.12 BLEU points compared to random training set. In translation tasks involving English and Turkish, a morphologically rich language, FDA5 can gain up to 11.52 BLEU points compared to a randomly selected subset of the same size, can achieve the same BLEU score using as little as 4% of the data compared to random instance selection, and can exceed the full dataset result by 0.78 BLEU.**

***Index Terms*—instance selection; machine translation; transductive learning; information retrieval**

EDICS Category: SPE-LANG

I. Introduction

STATISTICAL machine translation (SMT) makes use of a large number of parallel training sentences, which contain pairs of sentences that are translations of each other, to derive translation tables, estimate parameters, and generate the actual translation. Not all of the parallel training sentences 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 [1].

Previous work shows that the more the training data, the better the translations become [2]. Word-level translation accuracy is affected by the number of times a word occurs in the parallel training sentences [3]. 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 the diversity of the training set.

However, it is also common knowledge that the quality and the relevance of the training data have a significant impact on translation performance. With the increased size of the parallel training sentences 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 training sentences. Proper instance selection can play an important role in obtaining a small sized training set with which correct alignments can be learned. In this work, we quantify the effect of training data relevance and show that by using significantly less training data, we can achieve the same, or in some settings, higher level of translation performance.

Instance selection has been used in statistical machine translation in two ways:

Transductive learning makes use of test instances, which can sometimes be accessible at training time, to learn specific models tailored towards the test set. In a transductive learning setting, selection of training instances relevant to the test set improves the final translation quality [4].

Active learning selects a subset of training samples $\mathcal{L}$ from the unlabeled dataset $\mathcal{U}$ that will benefit a learning algorithm the most [5]. Active learning in SMT selects which instances to add to the training set to improve the performance of a baseline system [6] or which to retain for achieving similar performance using fewer instances [7], [8].

We describe a class of instance selection algorithms that use feature decay, feature decay algorithms (FDA), that aim to maximize the coverage of the target language features while decaying their weights and achieve significant gains in machine translation performance and decrease the training set size. FDA is introduced in [9] and in this paper, we develop FDA5, which is an independent extension of FDA that generalizes the ideas in earlier work with five parameters that allows better scaling, scoring, and optimization, which improves the overall performance and provides greater understanding of the domains and tasks together with identification of key differences between them. The parameterization and optimization mechanisms we introduce with FDA5 allows efficient instance selection with many monolingual and bilingual application scenarios. FDA5 can be used in both transductive and active learning scenarios. From a transductive learning perspective, we show that FDA5 can gain up to 3.22 BLEU points compared to a similarly sized randomly selected subset of the training set in an in-domain translation task with large parallel corpora and 11.52 BLEU points in a translation task involving English and Turkish, a morphologically rich language, with smaller parallel corpora. At the same time, FDA5 can also gain up to 0.43 BLEU points compared to using all of the available training data and can reach within 0.5 BLEU by using only 2.7% of the available training data. From an active learning perspective, we show that an SMT system

using FDA5 can achieve a given BLEU performance with as little as 4% of the available training data compared to random instance selection, significantly reducing the required human effort. In active learning experiments, FDA gains up to 0.45 BLEU points compared to using all of the available training data and 1.12 BLEU points compared to random training set. An implementation of the algorithm is available from the authors' website at `http://xxx.xxx.xxx`, which also includes a program for optimizing the parameters of FDA5.

The next section describes the general structure of feature decay algorithms. Section III describes related approaches to instance selection, some recast as specific instantiations of the FDA framework. We present a 5 parameter variation of FDA called FDA5 in Section IV and discuss its computational complexity. Section V presents our datasets, evaluation, optimization, and coverage results together with adaptation to in-domain (ID) and out-of-domain (OOD) translation tasks for different language pairs (English-German and English-Turkish). Section VI presents our translation results and Section VII presents parallel FDA5 algorithm. We summarize our contributions in the last section.

Algorithm 1: The Feature Decay Algorithm

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

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

II. Instance Selection with Feature Decay

In this section we will describe a class of instance selection algorithms for machine translation that use feature decay, which increases the diversity of the training set by devaluing features (i.e. n-grams) that have already been included. After reviewing the state of the art in the field, we generalize the main ideas in a class of feature decay algorithms (FDA) which allow efficient implementation and parameter optimization. Our abstraction makes three components of such algorithms explicit permitting experimentation with their alternatives:

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

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. FDA does this by reducing the value of the features that are already included after picking each training sentence from the source language. Algorithm 1 gives the pseudo-code for FDA.

The inputs to the algorithm are the source language training sentences $\mathcal{U}$, the source language features of the test set $\mathcal{F}$, and the desired number of words N in the subset $\mathcal{L}$ of the training set output by the program. We use n-grams up to a specified n as features in our experiments.

The first `foreach` loop initializes the value of each test set feature using `init`(f) which can use the frequency, length and other attributes of the n-grams to determine the feature value.

The second `foreach` loop initializes the score for each candidate training sentence using `sentScore`(S). This function uses the length of the sentence and the values of its features to estimate the utility of adding it to the output. The sentences are then pushed onto a queue with their scores.

Finally the `while` loop outputs a subset of the training sentences $\mathcal{L}$ by picking candidate sentences with the highest scores until the desired number of words N is reached. This is done by popping the top scoring candidate sentence S from the queue at each iteration. After ensuring that S is the best candidate it is placed in $\mathcal{L}$ and the values of its features are decreased using `decay`(f).

Note that as we change the feature values, the sentence scores in the queue will no longer be correct. However they will still be valid upper bounds because the feature values only get smaller. We use an abstract data type called an *upper bound queue* (implemented using a binary heap) that maintains an *upper bound* on the actual values of its elements [10]. Each successive *pop* from an upper bound queue is not guaranteed to retrieve the element with the largest value, but the remaining elements are guaranteed to have values smaller than or equal to the upper bound of the next element.

We thus recalculate the score of each sentence popped in the `while` loop because the values of its features may have changed. We compare the recalculated score of S with the upper bound of the next best candidate. If the score of S is equal or better we are sure that it is the top candidate, in which

case we place S in our training set and decay the values of its features. Otherwise we push S back into the priority queue with its updated score.

FDA gives us a class of algorithms that use feature decay for instance selection. By using upper bound queues implemented as binary heaps, FDA offers a very fast implementation for different instance selection algorithms. In the next section, we define various other models by parameterizing its three functions `init`, `decay`, and `sentScore`. Making the parameterization explicit allows us to optimize the parameters to discover better performing variants specialized to specific translation tasks.

A. FDA Framework

Biçici and Yuret [9] discover the FDA algorithm for training instance selection for machine translation given a training set and a test set in a transductive learning framework (*hence* [TL]). Training sentences are scored as follows:

$$\begin{aligned} \texttt{decay}(f) &= \texttt{init}(f)(1 + C_{\mathcal{L}}) \\ \texttt{init}(f) &= 1 \\ \texttt{sentScore}(S) &= \sum_{f \in F(S)} \textit{fvalue}(f) \end{aligned} \quad (1)$$

FDA is not parameterized and therefore optimization is only done by trying different decaying or initialization functions. Since there is no normalization with the sentence lengths, FDA also tends to select longer sentences to maximize the TCOV, which can make the word alignment task harder. In Section IV, we alleviate these problems with the introduction of FDA5, which parameterizes the contribution of each of these factors into consideration when calculating the value of features and the scores for sentences. Parameterization allows better understanding of the translation domains and tasks, improves the performance by adapting to new problems, and gives more control over what kind of instances are to be selected for the training set.

FDA is applied on many learning tasks which require diverse and relevant retrieval of training instances. FDA is very useful for MT since coverage and diversity are both important for building high performance SMT systems and the coverage of target features is correlated with the translation performance [9]. Recently, parallel FDA significantly reduces the time to deploy accurate MT systems to half a day and still achieve state-of-the-art SMT performance [11]. The same work also shows that if parallel FDA is used for selecting instances for the language model (LM) corpus using the FDA selected training target side as the test set, the relevancy of the LM corpus selected can reach up to 86% reduction in the number of OOV tokens and up to 74% reduction in the perplexity. FDA score is also used as an indicator of the expected translation quality [12], [13] for quality estimation in translation. Referential translation machines use FDA during monolingual retrieval of reference training sentences for making semantic similarity judgments [14] and grading student answers [15].

III. Related Work and FDA

In this section, we review the state of the art in the field of instance selection for machine translation. We recast some algorithms in the FDA framework and describe their differences using the three functions `init`, `decay`, and `sentScore`. We also categorize the related work into transductive learning (TL) and active learning (AL) approaches as described in the introduction depending on their emphasis in the original publication. In Section IV, we introduce FDA5, a variant of the FDA algorithm with five parameters that generalize many of the ideas introduced in earlier work.

N-gram coverage [AL]: Eck et al. [7] reduce the training set size by selecting a smaller subset after sorting the training data using a scoring function (*hence* [AL]). They try to maximize n-gram feature coverage when selecting training instances:

$$\begin{aligned} \texttt{init}(f) &= C_{\mathcal{U}}(f) \\ \texttt{decay}(f) &= (C_{\mathcal{L}}(f) > 0 \ ? \ 0 : \texttt{init}(f)) \\ \texttt{sentScore}(S) &= \frac{1}{|S|} \sum_{f \in F(S)} \textit{fvalue}(f) \end{aligned} \quad (2)$$

$\texttt{sentScore}(S)$ scores sentence S, $F(S)$ gives the set of features found in S, $C_{\mathcal{U}}$ and $C_{\mathcal{L}}$ return the count of f in $\mathcal{U}$ and $\mathcal{L}$ respectively. The NGRAM scorer 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. They do not use the test set when selecting training instances but rather use previously selected training data to identify the covered n-gram features.

TF-IDF [TL]: Lü et al. [4] use TF-IDF (term frequency - inverse document frequency) information retrieval technique based cosine score to select a subset of the parallel training sentences close to the test set for SMT training (*hence* [TL]). 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 n-gram 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. TF-IDF does not involve decay of feature values. If $\mathcal{T}$ is the test set and $C_{\mathcal{T}}(f)$ is the count of feature f in the test set, TF-IDF instance selection can be described in FDA terms as:

$$\begin{aligned} \texttt{init}(f) &= C_{\mathcal{T}}(f) \log(|\mathcal{T}|/C_{\mathcal{T}}(f))^2 \\ \texttt{decay}(f) &= \texttt{init}(f) \text{ (no decay)} \\ \texttt{sentScore}(S) &= \frac{\sum_{f \in F(S)} \textit{fvalue}(f)}{[\sum_{f \in F(S)} \log(|\mathcal{T}|/C_{\mathcal{T}}(f))^2]^{1/2}} \end{aligned} \quad (3)$$

DWDS [AL]: Density weighted diversity sampling (DWDS) [8] selects sentences containing the n-gram features in the unlabeled dataset $\mathcal{U}$ while increasing the diversity among the sentences selected, $\mathcal{L}$ (labeled) to improve SMT performance (*hence* [AL]). 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 DWDS scores as:

$$\begin{aligned} d(S) &= \frac{\sum_{x \in F(S)} P_{\mathcal{U}}(x)\, e^{-\alpha C_{\mathcal{L}}(x)}}{|F(S)|} \\ u(S) &= \frac{\sum_{x \in F(S)} I(x \notin F(\mathcal{L}))}{|F(S)|} \\ \texttt{sentScore}(S) &= \frac{2\, d(S)\, u(S)}{d(S) + u(S)}, \end{aligned} \tag{4}$$

where $F(S)$ stores the features of S, $I(.)$ is an indicator function, and α 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. DWDS tries to select sentences containing similar features in $\mathcal{U}$ with high diversity. In their experiments, they selected 1000 training instances in each iteration and retrained.

Perplexity [AL]: Perplexity of the training instance as well as inter-SMT-system disagreement are also used to select training data for translation models [16]. The increased difficulty in translating a parallel sentence as measured by the disagreements among translations obtained by a committee of translation models or its novelty as found by the perplexity adds to its importance for improving the SMT model's performance (*hence* [AL]). 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. SMT performance improvements can be achieved by training over some initial parallel training data together with selected subsets of additional training data instead of training with all of the available training data. Moore and Lewis [17] select training data for language models (LM) using the difference of the cross-entropy of ID and OOD training data:

$$H_{\text{ID}}(s) - H_{\text{OOD}}(s).$$

Axelrod et al. [18] use a bilingual cross-entropy difference score for selecting training data for SMT:

$$\phi_{aml}(s,t) = H^{S}_{\text{ID}}(s) - H^{S}_{\text{OOD}}(s) + H^{T}_{\text{ID}}(t) - H^{T}_{\text{OOD}}(t), \tag{5}$$

where S, T stand for the source and target languages, and (s,t) is a training sentence pair being scored.

IV. The FDA5 Algorithm

In this section we introduce a five parameter instance selection algorithm called FDA5. Explicitly parameterizing the three FDA functions `init`, `decay`, and `sentScore` allows us to (1) efficiently replicate and generalize over some of the ideas from earlier work, (2) optimize the parameters for any new ID or OOD target translation domain to achieve better performance, (3) control the type of instances that are selected from the training data, and (4) understand the target translation domains and tasks better. An implementation of the algorithm is available from the authors' website at `http://xxx.xxx.xxx`.

The FDA5 `init` function, which computes the initial value of a feature f can be parameterized to take into account the number of tokens in the feature $|f|$, and its log inverse frequency using the parameters l and i respectively. Features that do not appear in the test set are considered to have zero value.

$$\texttt{init}(f) = \log(|\mathcal{U}|/C_{\mathcal{U}}(f))^{i}\, |f|^{l} \tag{6}$$

The FDA5 `decay` function, which is used to compute the reduced values of features after they have been included $C_{\mathcal{L}}$ times in the output $\mathcal{L}$ can implement polynomial or exponential decay using the parameters c and d:

$$\texttt{decay}(f) = \texttt{init}(f)(1 + C_{\mathcal{L}})^{-c} d^{C_{\mathcal{L}}} \tag{7}$$

The FDA5 `sentScore` function calculates the total score for a sentence as a sum of its feature values and can be scaled by a sentence-length factor using the parameter s:

$$\texttt{sentScore}(S) = \frac{1}{|S|^{s}} \sum_{f \in F(S)} \mathit{fvalue}(f) \tag{8}$$

These five parameters, together with the maximum feature n-gram length n, determine the value of each sentence and the instance selection behavior of FDA5. The default values $d = 1, c = s = i = l = 0$ give every feature the same value and perform no decay or scaling.

A. Computational Complexity

The computational complexity of FDA5 is in $O(mM \log M)$, where M is the number of instances in the training data and m is the number of instances selected with $m \ll M$, which is dominated by the while loop. However, the average number of iterations depends on the sentence scores and the more the weight of a feature is decayed, the less it effects the score and hence the ordering, which makes the diversity more important. The number of iterations is also effected by the parameter values, which effect the scores. We investigate the computational cost of FDA5 by the average number of iterations in the main loop. Figure 1 shows the number of times the while loop iterates with respect to the number of words already selected for OOD and ID. The number of iterations in the while loop converges to one per word for OOD and two per word for ID instance selection using optimized parameters.

V. Datasets, Evaluation, and Optimization

We present the experimental settings for our results in three parts: datasets, evaluation, and optimization. FDA5 parameter optimization converges to very different values for different language pairs and even for in-domain and out-of-domain translation tasks. Section V-A describes the datasets we use. BLEU is an expensive metric to judge the performance of a training set, therefore we use target language bigram coverage (TCOV) as an alternative metric in some experiments as described in Section V-B. Section V-C describes how we obtain the optimal parameters for FDA5 and analyzes the sensitivity of results to each parameter. Finally, Section V-D introduces genetic algorithms as an alternative optimization

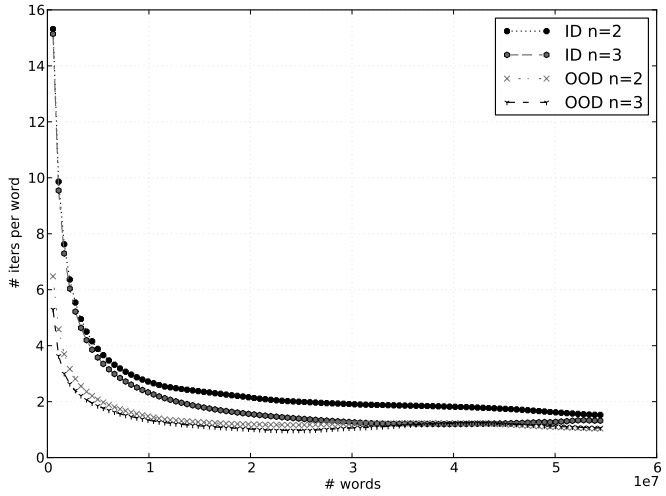

Fig. 1. Number of iterations in the while loop of FDA5 converges to one per word for OOD and two per word for ID instance selection.

method for searching for the parameters of FDA5, which reduces the computational overhead, and empirically achieves similar results. We use n-gram features.

A. Datasets

We performed optimization and sensitivity analysis for the parameters used in the FDA5 algorithm and obtained coverage results on the English (en) to German (de) language pair using the parallel training sentences provided by [19] (WMT'12). The English-German section of the Europarl corpus contains about 2 million sentences (55 million English, 52.5 million German words). Both the development set and the test set contain 3003 sentences (73K English, 72.6K German words) in this out-of-domain (OOD) translation task. We also created in-domain (ID) development and test sets composed of 1000 sentences (27K English, 26K German words) each by randomly sampling the training data. For ID experiments the development and test sets were removed from the training data. The target language training sets were used to build the language models required. We used the development sets to perform parameter optimization and sensitivity analysis and the test sets to perform feature coverage and BLEU evaluation. en-de language pair provides ID and OOD translation tasks with abundant and large parallel corpora.

Additionally, we perform optimization and obtain results on the English to Turkish (tr) and Turkish to English language pairs using the parallel training sentences provided by EU project Bologna [1], which contains course syllabi documentation from different universities in Turkey. The parallel corpus contains 352K training sentences (3.2 million English, 2.7 million Turkish words) and additional 1200 sentences each for development and test sets (14K English, 12K Turkish words). This language pair provides a translation task in a constrained domain with smaller parallel corpora and a harder one with Turkish being a morphologically rich language with scarce parallel corpora resources. The development and test sets are extracted randomly from the training set and hence this translation task is also in-domain.

[1] http://www.bologna-translation.eu/

B. Evaluation

Computing the BLEU score for each training set evaluated during optimization of instance selection is computationally expensive. Therefore we chose to use *target language bigram coverage* (TCOV) as a surrogate measure. TCOV measures the percentage of unique target language bigrams in the test/dev set included in a given training set. Note that FDA makes all instance selection decisions based on the source language and has no access to target language data. However the quality of the final translations depends on whether the correct target language phrases make it into the phrase table which motivates the TCOV measure.

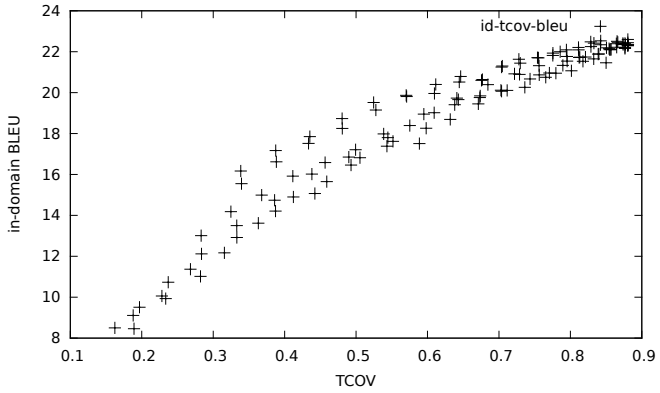

Fig. 2. Target language bigram coverage (TCOV) vs. BLEU scores from the in-domain experiments in this study showing the correlation between the two measures.

Figure 2 shows the empirical correlation between TCOV and BLEU on a scatter plot of a number of experiments we have performed in this study on in-domain datasets. The out-of-domain results are similar.

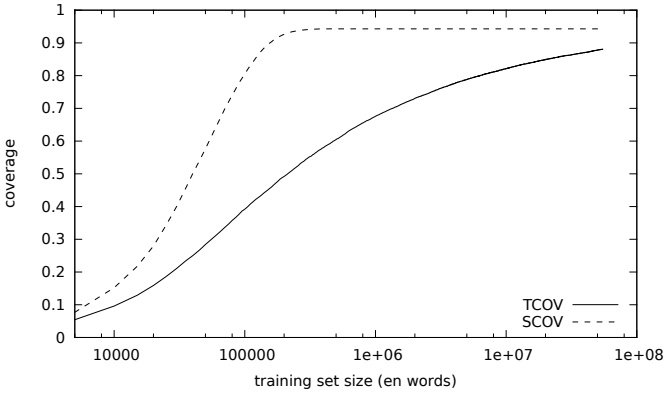

Fig. 3. Training set size vs. target language (TCOV) and source language (SCOV) bigram coverage for the optimized FDA5 instance selection on in-domain data.

Figure 3 shows the evolution of target and source language bigram coverage as more data is added to the training set by an optimized FDA5 algorithm on ID data. SCOV is maximized

out at 94.29% at around 0.5 million words of training data (less than 1% of the whole dataset). After this point there are no new source language features FDA5 can add to the dataset, but as new sentences are added, the *fvalue* for the same features are updated based on their initial value and the decay rate. As we can see, this continues to improve TCOV until it reaches 88.06% with the full dataset. For out-of-domain experiments the curves have a similar shape, reaching 74.52% SCOV and 64.37% TCOV with the full dataset.

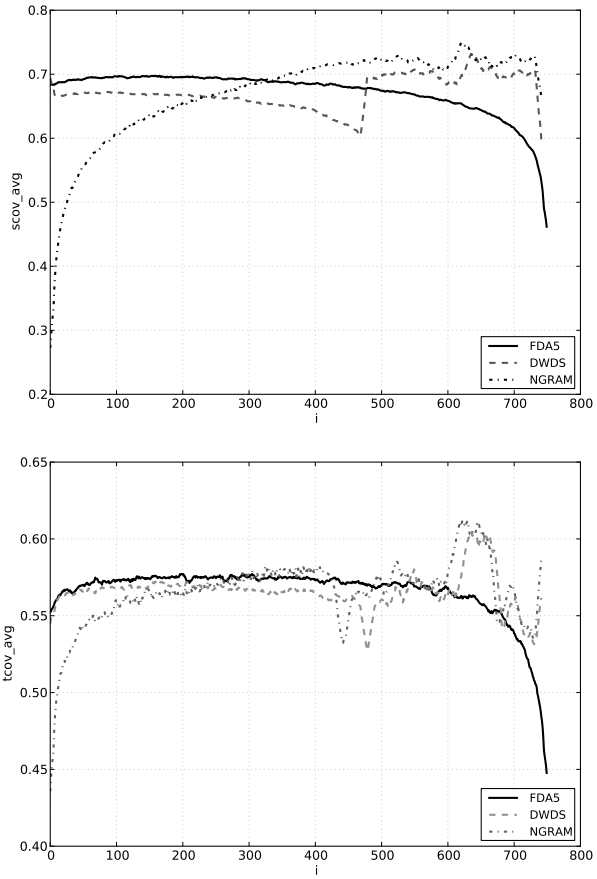

Fig. 4. Instance selection quality by the consistency of the newly selected training instances as measured by SCOV (top) and TCOV (bottom) for OOD. Average changes in SCOV and TCOV are depicted as instances are selected.

We measure the instance selection quality of the selection models as more instances are selected by the consistency of the SCOV and TCOV levels. Figure 4 measures the added value after each 73K source word additions (the size of the OOD test set) by looking at the relevancy and diversity as quantified by the SCOV and the TCOV obtained in an averaged window of 5 items for OOD experiments. We observe that FDA5 outperforms both DWDS and NGRAM by consistently selecting instances with high source and target coverage.

C. Optimal Parameters for FDA5

We searched the parameter space of FDA5 using a combination of grid search and the DHC optimization algorithm [20] to find values that optimized TCOV on the development set using 1 million words of training data. For in-domain data, we found an optimum at $d = 1$, $c = 2.296$, $s = 1.1$, $i = 0$, $l = 0$, $n = 3$ giving a TCOV value of 0.6731 and for out-of-domain, we found an optimum at $d = 1$, $c = 0.25$, $s = 0.8$, $i = 5.2552$, $l = -0.4$, $n = 2$ giving a TCOV value of 0.4196.

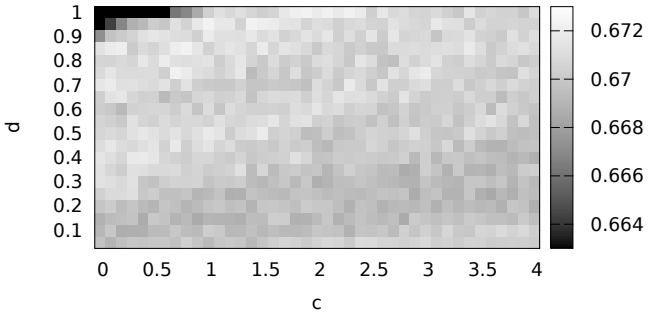

Fig. 5. c-d grid for in-domain data with shades of gray representing TCOV at 1M words with points not within 1% of the optimum value painted black. Other parameters are set to $n = 3, s = 1, i = l = 0$.

Early on we discovered that using trigrams ($n = 3$), as well as words and bigrams, benefits the ID results but not OOD results, even though in both cases we evaluate the output using TCOV which uses bigrams. Figure 5 shows that many combinations of the polynomial (c) and exponential (d) decay parameters give very similar results. With the exception of the black region at the upper left ($c = 0, d = 1$, no decay) all points in the grid are within 1% TCOV of the optimum.

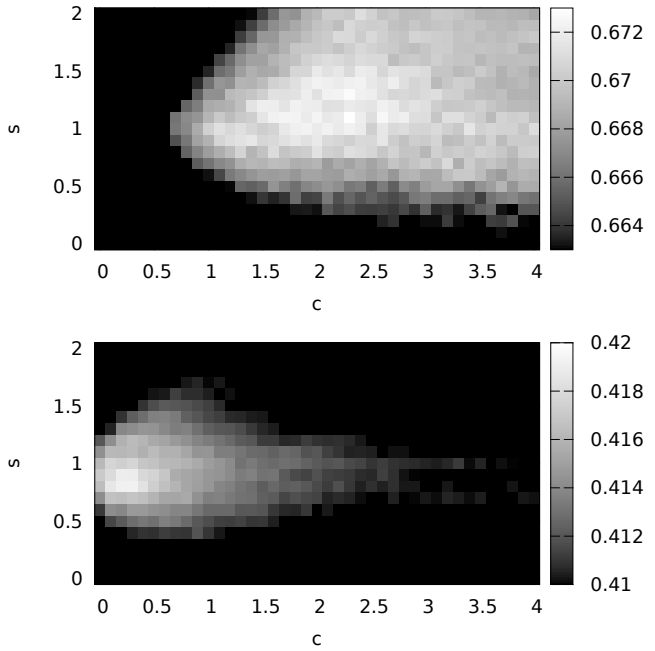

Fig. 6. c-s grids for ID (top) and OOD (bottom) datasets. Shades of gray represent TCOV at 1M words with points that are not within 1% of the optimum value painted black. Other parameters are set to $n = 3, d = 1, i = l = 0$ for ID and $n = 2, d = 1, i = 5.2552, l = -0.4$ for OOD.

Figure 6 shows that a larger decay rate is better for ID experiments compared to OOD experiments. As we see in Figure 7, OOD results are more sensitive to the initial values

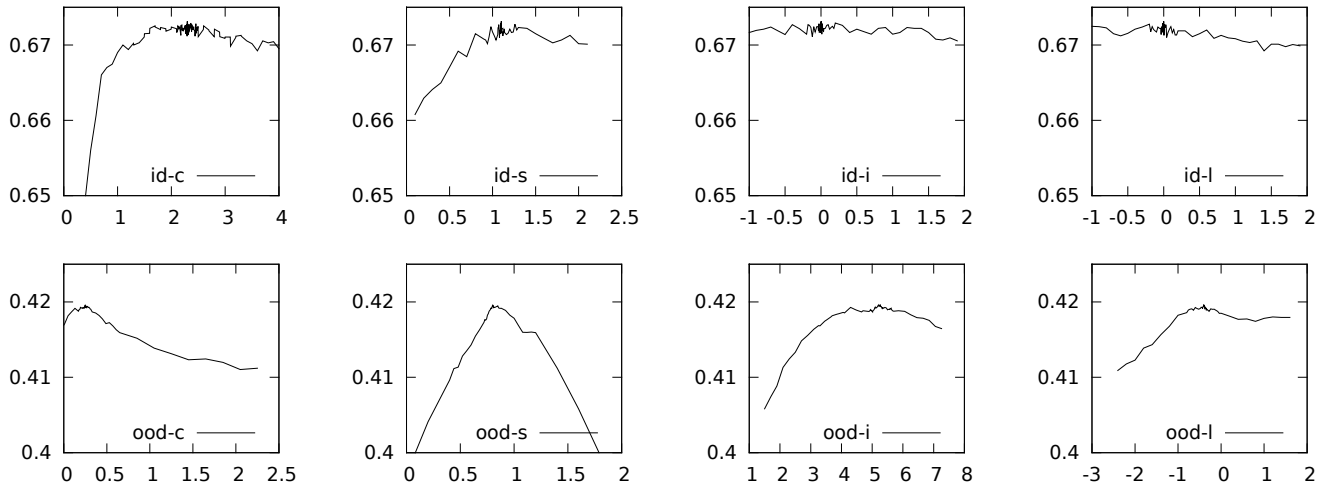

Fig. 7. Sensitivity of target language bigram coverage (y-axes) to changes in the parameters c, s, i, and l (x-axes). The first row shows results from in domain experiments (initial $c = 2.3, s = 1.1, i = l = 0$), the second row shows results from out of domain experiments (initial $c = 0.25, s = 0.8, i = 5.2552, l = -0.4$). The n-gram order is $n = 3$ for in domain, $n = 2$ for out of domain, and $d = 1$ (no exponential decay) for both sets of experiments.

of features (preferring shorter and less frequent features) and less on decay rate. In fact with no decay ID results get significantly worse, but OOD results stay within 1% of the optimum. Figure 6 also shows that a sentence normalization with $s \approx 1$ is necessary for both ID and OOD performance.

Figure 7 plots sensitivity of TCOV with respect to changes in the optimal parameter settings we learned. We observe several key differences between ID and OOD results:

- Longer features ($n = 3$) benefit ID more than OOD.
- Initial values (`init`) are important for OOD, which prefers short and infrequent features, but not for ID.
- A fast decay rate ($c > 1$) is crucial for ID, which falters with no decay, whereas a low decay ($c < 1$) is optimal for OOD, which does OK even with no decay ($c = 0$).
- Various combinations of exponential ($d < 1$) and polynomial ($c > 0$) decay give similar results, but at the end we found polynomial decay was slightly better.
- Sentence normalization ($s \approx 1$) is important for ID but more so for OOD.

D. Optimization with Genetic Algorithms

Searching the parameter space of FDA5 requires a combination of computationally expenvise grid search and several DHC optimization steps to be run, requiring several days to be spent for finding optimal parameters for a given N, the number of words. This section introduces an alternative method, evolution strategy (ES) for optimization, which can find the optimal or very close to the optimal solution for this complex optimization problem in the order of hours. Evolution strategy [21] is a variant of genetic algorithms where real valued parameter populations evolve towards the optimal solution after several generations of mutations.

By using ES, we can empirically obtain good results in a couple of hours, which allows us to perform optimization for any given N, the desired number of training words. Figure 8 plots the changes in the parameter values as the parameters

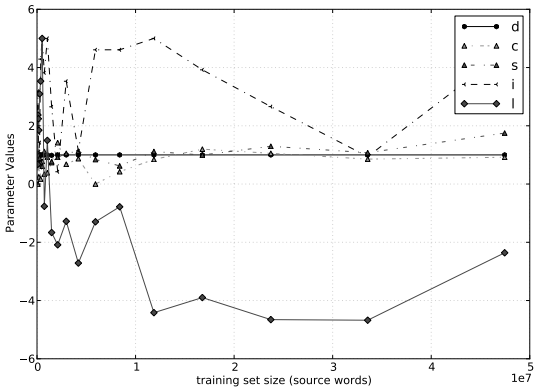

Fig. 8. Changes in parameter values optimized with ES for OOD with increasing training set size.

of FDA5 is optimized with ES for increasing N for the OOD translation task. We select the parameters with n for which the optimization leads to higher TCOV value. ES finds very close parameters to the parameters we found for 1M words using DHC and grid search in the previous section: $d = 1.0$, $c = 0.387$, $s = 0.9251$, $i = 5.0$, and $l = 1.498$. As the training set size increase, optimal value for l decrease and i increase showing a preference towards including longer and rarer features. d, c, and s vary around 1 with d and c being closely related yet both with positive values, showing that both exponential and polynomial decay are important for better selection of the training set. The mean values for the parameters after optimization are given in Table I. Most translation tasks prefer $n = 3$ more than $n = 2$ except for OOD. OOD prefers more exponential decay and less polynomial decay than others and shortest and rarest features. For active learning experiments (Section VI-C), we obtain the largest sentence and feature length and log inverse penalties.

μ	n	d	c	s	i	l
en-de (ID)	2.90	0.932	1.607	1.033	1.882	-2.617
en-de (OOD)	2.35	0.968	0.729	0.961	3.073	-0.517
en-tr	3.00	0.870	1.844	0.962	2.072	-2.038
tr-en	2.91	0.455	1.992	0.123	2.584	-3.025
en-de (ID AL)	2.55	0.658	1.086	1.037	3.338	1.310
en-de (OOD AL)	2.6	0.767	0.977	1.020	3.427	0.980

TABLE I
MEAN PARAMETER VALUES FOR DIFFERENT TRANSLATION TASKS.

VI. MACHINE TRANSLATION PERFORMANCE

In this section, we provide a comparison of FDA5 machine translation performance with related work in English-German (Section VI-A), English-Turkish (en-tr), and Turkish-English (tr-en) (Section VI-B) translation tasks. We compare FDA5 with a random instance selection baseline and other related methods in terms of the BLEU score. We optimize FDA5 parameters for each N with evolution strategy, which turns out to achieve close performance to the full optimization with grid search and DHC. The baseline performance using all of the available training corpora are 22.55 BLEU for ID and 13.82 BLEU for OOD translation tasks and 24.45 BLEU for en-tr and 29.61 BLEU for tr-en translation tasks.

	BLEU gain		data ratio	% of data for BLEU-0.5
wrt.	RAND	ALL	RAND	ALL
ID	$+3.22$	$+0.01$	1/8	11%
OOD	$+2.09$	$+0.43$	1/11.3	2.7%
en-tr	$+11.23$	$+0.78$	1/23	8%
tr-en	$+11.52$	$+0.0$	1/23	19%
ID AL	$+0.38$	$+0.0$	1/2	43%
OOD AL	$+1.12$	$+0.45$	1/6	5%

TABLE II
SUMMARY OF FDA5'S TRANSLATION PERFORMANCE. POSSIBLE BLEU GAINS WITH RESPECT TO USING ALL OF THE TRAINING DATA (ALL) OR TO RANDOM BASELINE (RAND) ARE GIVEN IN THE FIRST TWO COLUMNS. THE NEXT COLUMN LIST THE RATIO OF THE FDA5 TRAINING DATA TO RAND TRAINING DATA TO REACH THE SAME BLEU PERFORMANCE. THE LAST COLUMN IS THE PERCENTAGE OF ALL THE TRAINING DATA REQUIRED FOR REACHING WITHIN 0.5 BLEU TO ALL PERFORMANCE.

As we demonstrate in the following subsections, FDA5 achieves significant gains in the translation performance. The summary of FDA5's translation results are given in Table II. FDA5 can gain up to 11.52 BLEU points compared to a randomly selected training set of the same size, or achieve similar BLEU performance using up to 23 times less data. FDA5 can also gain up to 0.43 BLEU points compared to using all of the available training data and can reach within 0.5 BLEU by using only 2.7% of the available training data for OOD translation. The gains reach 0.78 BLEU points for the en→tr translation task. Larger BLEU gains and smaller selected training data for reaching high BLEU scores in the OOD and en→tr translation tasks with Turkish being a higher vocabulary language, indicate that FDA5 performs especially well in harder translation tasks. In active learning experiments, FDA gains up to 0.45 BLEU points compared to using all of the available training data and 1.12 BLEU points compared to random training set.

A. English-German Results

We obtained translation results on the English (en) to German (de) language pair using the parallel training sentences as described in Section V-A. Figure 9 compares the optimized FDA5 instance selection with a random instance selection baseline and other instance selection methods for a range of training set sizes in terms of BLEU score for ID and OOD experiments. The first figure gives training set size vs BLEU for the 27K word in-domain test set where the training data is selected from the 55M word WMT12 en→de parallel training set (filtered to exclude the dev and test sentences). The second figure presents a similar comparison for the official 73K word out-of-domain test data and subsets of the WMT12 en→de training set.

FDA5 optimized for in-domain data (the top line labeled FDA5) gains up to 3.22 BLEU points compared to a randomly selected training set (line with labeled RAND) of the same size, or to reach the same BLEU performance as FDA5, random instance selection needs up to 8 times more data. FDA5 optimized for out-of-domain data (the top line labeled FDA5 on the right figure) gains up to 2.09 BLEU points compared to a randomly selected training set (line labeled RAND) of the same size, or to reach the same BLEU performance as FDA5, random instance selection needs up to 11.3 times more data.

All other methods with the exception of DWDS give performances significantly below FDA5, and in the case of in-domain data, even below random instance selection for small training sets. Optimized FDA5 outperforms DWDS in both the in-domain experiments (up to 0.37 BLEU points) and in the out-of-domain experiments (up to 0.35 BLEU points). These results indicate that methods that do not use exponential feature decay or that do not take into account the test set features such as NGRAM do not perform as well as the ones that do.

	ID			OOD		
Model	bigrams	wps	TCOV	bigrams	wps	TCOV
FDA5	346K	19	.68	426K	24	.42
DWDS	351K	20	.67	412K	19	.42
NGRAM	517K	21	.57	514K	17	.37
RAND	349K	25	.61	347K	25	.34

TABLE III
STATISTICS OF THE TARGET $\mathcal{L}$ FOR ID AND OOD TEST SETS USING 10^6 TARGET WORDS. BIGRAMS LIST THE UNIQUE 2-GRAMS FOUND AND WPS IS THE NUMBER OF WORDS PER SENTENCE.

The statistics of $\mathcal{L}$ obtained with the instance selection techniques differ from each other as given in Table III, where 10^6 source training words are selected for ID and OOD test sets. FDA5 achieves top coverage along with DWDS and achieves better TCOV using fewer unique bigrams in ID. NGRAM is not able to discriminate between sentences well and a large number of sentences of the same length get the same score when the unseen n-grams belong to the same frequency class. NGRAM obtains the largest number of unique target bigrams.

Both FDA5 and other instance selection methods converge to the same BLEU result at the end when using the full 55M word training set. However FDA5 reaches within 0.5 BLEU

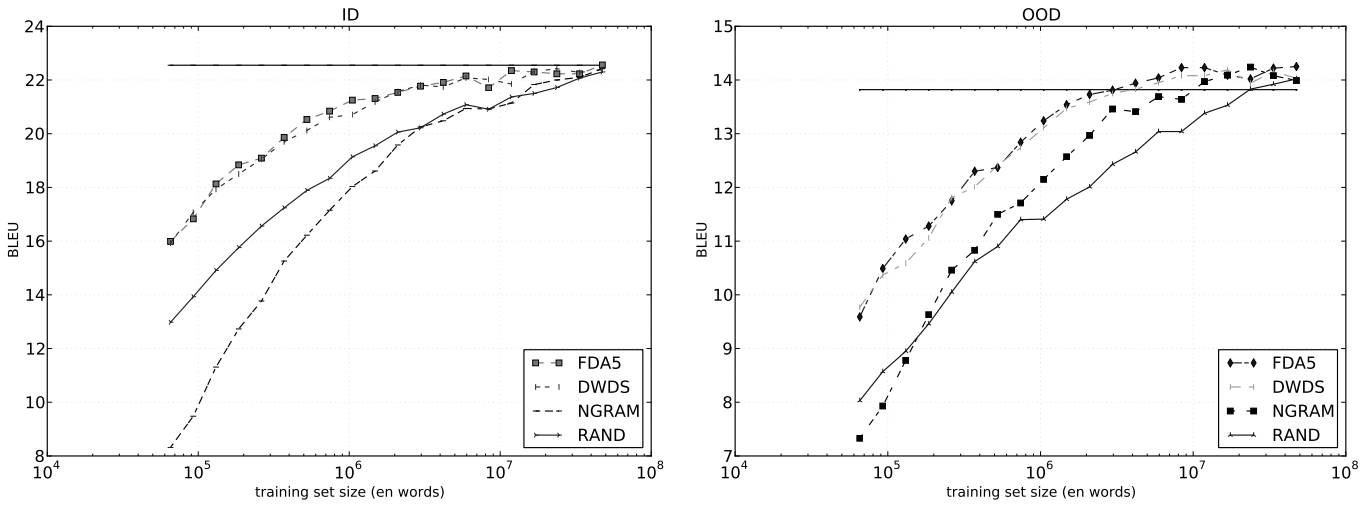

Fig. 9. A comparison of optimized FDA5 with baseline random instance selection and other related methods (straight line corresponds to the BLEU using all of the training set). The first figure gives training set size vs. BLEU for ID experiments and the second figure gives the results for OOD experiments.

of this result using less than 11% of the data for in-domain and less than 2.7% of the data for out-of-domain data. FDA5 peaks at around 12M words or 20% of the full training set, for both sets of experiments exceeding the full dataset result by 0.43 BLEU for out-of-domain data.

B. English-Turkish and Turkish-English Results

We obtained translation results on the English (en) to Turkish (tr) language pair using the parallel training sentences as described in Section V-A. Figure 10 compares the optimized FDA5 instance selection with a random instance selection baseline for a range of training set sizes in terms of BLEU score. The first figure gives results in the en→tr translation task and the second one in the tr→en translation task.

In the en→tr translation task, FDA5 gains up to 11.23 BLEU points compared to a randomly selected training set of the same size, or to reach the same BLEU performance as FDA5, random instance selection needs up to 23 times more data. In tr→en direction, FDA5 gains up to 11.52 BLEU points compared to a randomly selected training set of the same size, or to reach the same BLEU performance as FDA5, random instance selection again needs up to 23 times more data.

However FDA5 reaches within 0.5 BLEU to the BLEU result obtained using the full training set using about 8% of the data for en→tr and about 19% of the data for tr→en. FDA5 exceeds the full dataset result by 0.78 BLEU for en→tr.

C. Active Learning Results

We obtained translation results when using FDA5 in an active learning setting where we use the training set features as the test set features for selecting training instances. Figure 11 compares the FDA5 instance selection optimized according to the training set with a random instance selection baseline for a range of training set sizes in terms of BLEU score for ID AL and OOD AL translation tasks. FDA in OOD AL gains up to 0.45 BLEU points compared to using all of the training data and 1.12 BLEU points compared to random training set.

Algorithm 2: Parallel FDA5

Input: $\mathcal{U}$, $\mathcal{F}$, and N.
Output: $\mathcal{L} \subseteq \mathcal{U}$.

$\mathcal{U} \leftarrow \texttt{shuffle}(\mathcal{U})$
$\boldsymbol{\mathcal{U}}, M \leftarrow \texttt{split}(\mathcal{U}, N)$
$\boldsymbol{\mathcal{L}} \leftarrow \{\}$; $\boldsymbol{\mathcal{S}} \leftarrow \{\}$
foreach $\mathcal{U}_i \in \boldsymbol{\mathcal{U}}$ **do**
  $\mathcal{L}_i, \mathcal{S}_i \leftarrow \texttt{FDA5}(\mathcal{U}_i, \mathcal{F}, M)$
  $\texttt{add}(\boldsymbol{\mathcal{L}}, \mathcal{L}_i)$
  $\texttt{add}(\boldsymbol{\mathcal{S}}, \mathcal{S}_i)$
$\mathcal{L} \leftarrow \texttt{merge}(\boldsymbol{\mathcal{L}}, \boldsymbol{\mathcal{S}})$

Previous work on AL could not achieve better results than baseline system results [7] whereas our results show that better BLEU results are possible with using FDA5 in AL setting for OOD translation task.

VII. Parallel FDA5

FDA5 obtains a sorting of the training instances according to the weights of the test set features. Any change in the instance selection order results with a new scoring and ordering of the instances, making parallelization of the FDA5 algorithm difficult; but we can follow the approach in [11] to improve the scalability and the diversity further. Parallel FDA5 (Algorithm VII) first shuffles the training sentences, $\mathcal{U}$ and runs individual FDA5 models on the multiple splits from which equal number of sentences, M, are selected. $\texttt{merge}$ combines k sorted lists, $\mathcal{L}_i$, into one sorted list in $O(Mk \log k)$ using their scores, $\mathcal{S}_i$, where Mk is the total number of elements in all of the input lists. [2] Parallel FDA5 achieves close performance to FDA5 in terms of the target 2-gram feature coverage. Parallel FDA5 makes FDA5 more scalable to domains with large training corpora and allows

[2] [22], question 6.5-9. Merging k sorted lists into one sorted list using a min-heap for k-way merging.

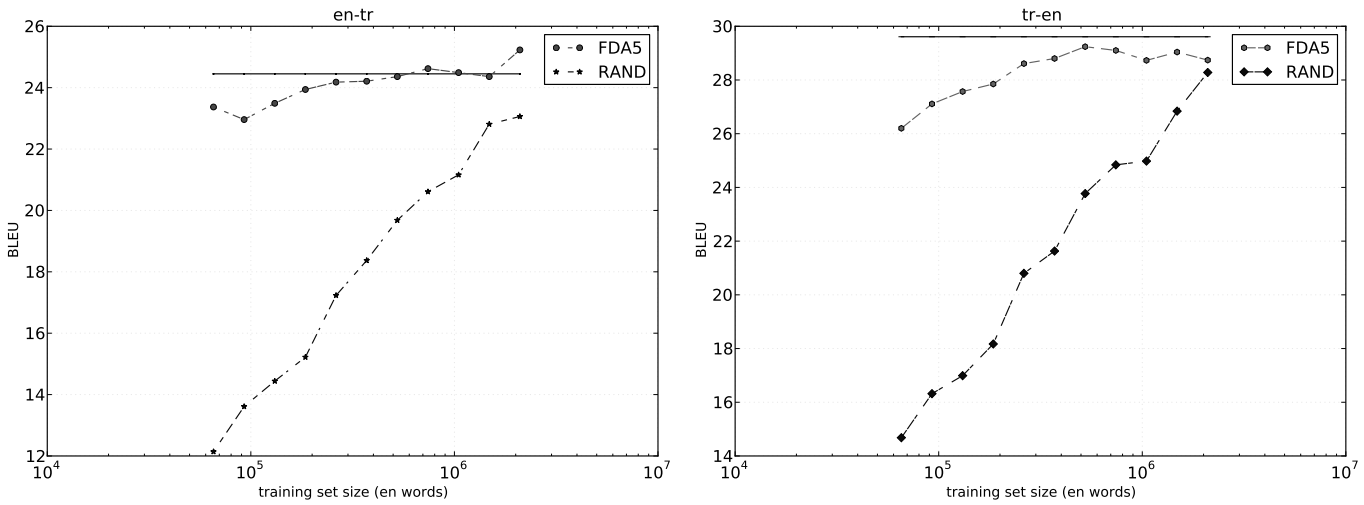

Fig. 10. A comparison of optimized FDA5 with baseline random instance selection (straight line corresponds to the BLEU using all of the training set). The first figure gives training set size vs. BLEU for en-tr experiments and the second figure gives the results for tr-en experiments.

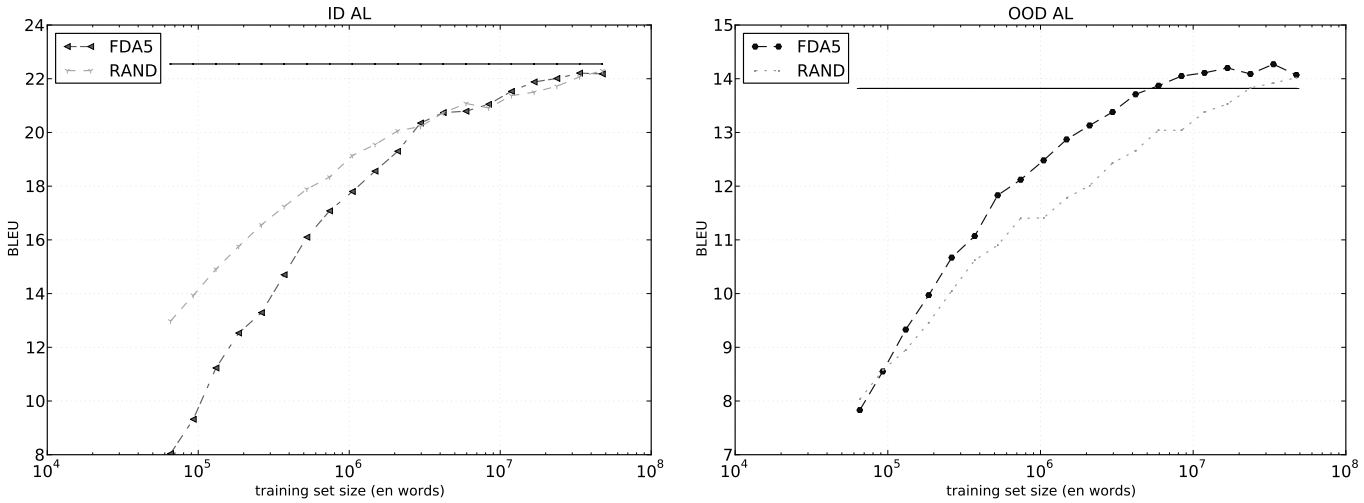

Fig. 11. A comparison of optimized FDA5 in active learning setting with baseline random instance selection (straight line corresponds to the BLEU using all of the training set). The first figure gives training set size vs. BLEU for ID AL experiments and the second figure gives the results for OOD AL experiments.

rapid deployment of SMT systems. By selecting from random splits of the original corpus, we work with different n-gram feature distributions in each split and prevent weights become negligible, which can enhance the diversity.

VIII. Contributions

We have introduced feature decay algorithms (FDA), a class of instance selection algorithms for machine translation that use feature decay, which generalize some of the ideas from related work, and allow optimization and efficient implementation. We describe some of the best performing instance selection algorithms as special cases of FDA.

We define a 5 parameter FDA instantiation called FDA5, and optimized its parameters on in-domain and out-of-domain translation tasks in different language pairs showing that different feature values and decay rates are appropriate for different tasks. We use target language bigram coverage (TCOV) for evaluation during optimization for efficiency and show that it correlates well with BLEU. We show that the average amount of exponential and polynomial decaying we perform with the optimal parameters are the same for translating from English to German and very close to the amount for translating from English to Turkish. The average amount of decaying and scaling is less when translating from Turkish to English where much longer and more common features are prefered.

FDA5 significantly outperforms other instance selection methods we have implemented except to a lesser degree for DWDS, which is another special case of FDA. A comparison with random instance selection shows that FDA5 can gain up to 3.22 BLEU points for English-German and up to 11.52 BLEU points for English-Turkish translation tasks at the same training set size achieving significant performance improvement, or can achieve a comparable BLEU result using as little as 4% of the data achieving significant reductions in the training set size. In the English-German translation tasks we have tested, FDA5 performance peaks at less than

20% of the training set exceeding the result with the full training set by 0.43 BLEU for out-of-domain test set and can reach within 0.5 BLEU by using only 2.7% of the available training data. Also, in the English to Turkish translation task, FDA5 performance exceeds the result with the full training set by 0.78 BLEU. These results show that a smaller but more relevant subset of the training set can give us better accuracy in statistical machine translation. An implementation of the algorithm is available from the authors' website at `http://xxx.xxx.xxx`, which also includes a program for optimizing the parameters of FDA5.

REFERENCES

[1] 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, "Moses: Open source toolkit for statistical machine translation," in *Annual Meeting of the Assoc. for Computational Linguistics*, Prague, Czech Republic, Jun. 2007, pp. 177–180.

[2] P. Koehn, "Statistical machine translation: the basic, the novel, and the speculative," 2006, tutorial at EACL 2006.

[3] P. Koehn and K. Knight, "Knowledge sources for word-level translation models," in *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, 2001.

[4] Y. Lü, J. Huang, and Q. Liu, "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)*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 343–350. [Online]. Available: http://www.aclweb.org/anthology/D/D07/D07-1036

[5] M. Banko and E. Brill, "Scaling to very very large corpora for natural language disambiguation," in *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*. Toulouse, France: Association for Computational Linguistics, July 2001, pp. 26–33. [Online]. Available: http://www.aclweb.org/anthology/P01-1005

[6] G. Haffari, M. Roy, and A. Sarkar, "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*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 415–423. [Online]. Available: http://www.aclweb.org/anthology/N/N09/N09-1047

[7] M. Eck, S. Vogel, and A. Waibel, "Low cost portability for statistical machine translation based on n-gram coverage," in *Proceedings of the 10th Machine Translation Summit, MT Summit X*, Phuket, Thailand, September 2005, pp. 227–234.

[8] V. Ambati, S. Vogel, and J. Carbonell, "Active learning and crowd-sourcing for machine translation," in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, N. C. C. Chair), K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, Eds. Valletta, Malta: European Language Resources Association (ELRA), May 2010.

[9] E. Biçici and D. Yuret, "Instance selection for machine translation using feature decay algorithms," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, July 2011, pp. 272–283. [Online]. Available: http://www.aclweb.org/anthology/W11-2131

[10] D. Yuret, "FASTSUBS: An efficient and exact procedure for finding the most likely lexical substitutes based on an n-gram language model," *Signal Processing Letters, IEEE*, vol. 19, no. 11, pp. 725–728, Nov 2012.

[11] E. Biçici, "Feature decay algorithms for fast deployment of accurate statistical machine translation systems," in *Proceedings of the Eigth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013.

[12] E. Biçici, D. Groves, and J. van Genabith, "Predicting sentence translation quality using extrinsic and language independent features," *Machine Translation*, 2013.

[13] E. Biçici, "Referential translation machines for quality estimation," in *Proceedings of the Eigth Workshop on Statistical Machine Translation*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013.

[14] E. Biçici and J. van Genabith, "CNGL-CORE: Referential translation machines for measuring semantic similarity," in **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*. Atlanta, Georgia, USA: Association for Computational Linguistics, 13-14 June 2013.

[15] ——, "CNGL: Grading student answers by acts of translation," in **SEM 2013: The Second Joint Conference on Lexical and Computational Semantics and Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, 14-15 June 2013.

[16] A. Mandal, D. Vergyri, W. Wang, J. Zheng, A. Stolcke, G. Tur, D. Hakkani-Tur, and N. Ayan, "Efficient data selection for machine translation," in *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE*, Dec 2008, pp. 261 –264.

[17] R. C. Moore and W. Lewis, "Intelligent selection of language model training data," in *ACL (Short Papers)*, 2010, pp. 220–224.

[18] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *EMNLP*, 2011, pp. 355–362.

[19] C. Callison-Burch, P. Koehn, C. Monz, M. Post, R. Soricut, and L. Specia, "Findings of the 2012 workshop on statistical machine translation," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, June 2012, pp. 10–51. [Online]. Available: http://www.aclweb.org/anthology/W12-3102

[20] D. Yuret, "From genetic algorithms to efficient optimization," MIT AI Laboratory, Tech. Rep. 1569, 1994.

[21] K. A. D. Jong, *Evolutionary computation - a unified approach*. MIT Press, 2006.

[22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.