Instance Selection for Machine Translation using Feature Decay Algorithms

Ergun Biçici
ebicici@ku.edu.tr

Deniz Yuret
dyuret@ku.edu.tr

Department of Computer Engineering
Koç University, Istanbul, Turkey

WMT'11, Edinburgh, England

July 31, 2011

OUTLINE

1. INSTANCE SELECTION FOR MACHINE TRANSLATION
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU

Outline

1. Instance Selection for Machine Translation
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU
2. Experimental Results
   - *tcov* Comparison
   - Translation Results
   - *dice*: Instance Selection for Alignment

OUTLINE

1. INSTANCE SELECTION FOR MACHINE TRANSLATION
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU
2. EXPERIMENTAL RESULTS
   - *tcov* Comparison
   - Translation Results
   - *dice*: Instance Selection for Alignment
3. CONTRIBUTIONS

OUTLINE

1. INSTANCE SELECTION FOR MACHINE TRANSLATION
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU
2. EXPERIMENTAL RESULTS
   - *tcov* Comparison
   - Translation Results
   - *dice*: Instance Selection for Alignment
3. CONTRIBUTIONS

Instance Selection for Machine Translation

- We perform an empirical study of instance selection techniques for machine translation.
- Proper instance selection plays an important role in obtaining a small sized training set with which correct alignments can be learned.
- Previous work show that:
    - The more the training data, the better the translations become [Koehn, 2006]. (doubling training data size improves BLEU score by 1, doubling LM data by 0.5)
    - Word-level translation accuracy is affected by the number of times a word occurs in the parallel corpus [Koehn and Knight, 2001].
- Feature decay algorithms (FDAs) increase diversity of the training set by devaluing features that are already included.

INSTANCE SELECTION FOR MACHINE TRANSLATION

- FDAs optimize the source coverage weighted by decreasing feature weights
- FDAs try to select few instances for maximum coverage.
- We show that (using Moses):
  - High coverage corresponds to high BLEU score.
  - 3000 training sentences for a specific test sentence is sufficient to obtain a score within 1 BLEU of the baseline.
  - 5% of the training data is sufficient to exceed the baseline.
  - $\sim$ 2 BLEU improvement over the baseline is possible by optimally selected subset (20%) of the training data.
  - 7% of the training data is enough to achieve a similar performance with the baseline in out-of-domain translation.

Outline

1. Instance Selection for Machine Translation
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU

2. Experimental Results
   - *tcov* Comparison
   - Translation Results
   - *dice*: Instance Selection for Alignment

3. Contributions

Related Work

- Previous work in regression-based machine translation selects instances per sentence using the *tf-idf* metric or per feature.
- Active learning (AL) vs. Transductive Learning (TL) examples:
  - TFIDF (TL): [Lü et al., 2007] use tf-idf to select training instances.
  - NGRAM (AL): [Eck et al., 2005] use *n*-gram coverage.
  - DWDS (AL): [Ambati et al., 2010] use *n*-gram densities and diversities to select.
  - ELPR (AL): [Haffari and Sarkar, 2009] use *n*-gram frequency ratios to select.

Outline

1. Instance Selection for Machine Translation
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU
2. Experimental Results
   - *tcov* Comparison
   - Translation Results
   - *dice*: Instance Selection for Alignment
3. Contributions

Feature Decay Algorithms

- We show 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.
- We seek to maximize the coverage or the percentage of test source and target features found in the training set using minimal number of target training features and a fixed number of training instances.
- Features can be single words, bigrams, or phrases
- A word not found in the training set is impossible to translate
- Multiple translations exist; covering a source feature does not necessarily mean covering the target feature
- Feature Decay Algorithm (FDA) tries to increase the chance of covering the target test features by decreasing the weight of covered source features.

Feature Decay Algorithm

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

foreach $\underline{f \in \mathcal{F}}$ **do**
 $\texttt{fvalue}(f) \leftarrow \texttt{init}(f, \mathcal{U})$
foreach $\underline{S \in \mathcal{U}}$ **do**
 $\texttt{score}(S) \leftarrow \sum_{f \in \texttt{features}(S)} \texttt{fvalue}(f)$
 $\texttt{push}(\mathcal{Q}, S, \texttt{score}(S))$
while $\underline{|\mathcal{L}| < N}$ **do**
 $S \leftarrow \texttt{pop}(\mathcal{Q})$
 $\texttt{score}(S) \leftarrow \sum_{f \in \texttt{features}(S)} \texttt{fvalue}(f)$
 if $\underline{\texttt{score}(S) \geq \texttt{topval}(\mathcal{Q})}$ **then**
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{S\}$
  foreach $\underline{f \in \texttt{features}(S)}$ **do**
   $\texttt{fvalue}(f) \leftarrow \texttt{decay}(f, \mathcal{U}, \mathcal{L})$
 else
  $\texttt{push}(\mathcal{Q}, S, \texttt{score}(S))$

Feature Decay Algorithm

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

foreach $f \in \mathcal{F}$ **do**
    $\texttt{fvalue}(f) \leftarrow \texttt{init}(f, \mathcal{U})$
foreach $S \in \mathcal{U}$ **do**
    $\texttt{score}(S) \leftarrow \sum_{f \in \texttt{features}(S)} \texttt{fvalue}(f)$
    $\texttt{push}(\mathcal{Q}, S, \texttt{score}(S))$
while $|\mathcal{L}| < N$ **do**
    $S \leftarrow \texttt{pop}(\mathcal{Q})$
    $\texttt{score}(S) \leftarrow \sum_{f \in \texttt{features}(S)} \texttt{fvalue}(f)$
    if $\texttt{score}(S) \geq \texttt{topval}(\mathcal{Q})$ **then**
        $\mathcal{L} \leftarrow \mathcal{L} \cup \{S\}$
        foreach $f \in \texttt{features}(S)$ **do**
            $\texttt{fvalue}(f) \leftarrow \texttt{decay}(f, \mathcal{U}, \mathcal{L})$
    else
        $\texttt{push}(\mathcal{Q}, S, \texttt{score}(S))$

Feature Decay Algorithm

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

foreach $\underline{f \in \mathcal{F}}$ **do**
 $\texttt{fvalue}(f) \leftarrow \texttt{init}(f, \mathcal{U})$
foreach $\underline{S \in \mathcal{U}}$ **do**
 $\texttt{score}(S) \leftarrow \sum_{f \in \texttt{features}(S)} \texttt{fvalue}(f)$
 $\texttt{push}(\mathcal{Q}, S, \texttt{score}(S))$
while $\underline{|\mathcal{L}| < N}$ **do**
 $S \leftarrow \texttt{pop}(\mathcal{Q})$
 $\texttt{score}(S) \leftarrow \sum_{f \in \texttt{features}(S)} \texttt{fvalue}(f)$
 if $\underline{\texttt{score}(S) \geq \texttt{topval}(\mathcal{Q})}$ **then**
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{S\}$
  foreach $\underline{f \in \texttt{features}(S)}$ **do**
   $\texttt{fvalue}(f) \leftarrow \texttt{decay}(f, \mathcal{U}, \mathcal{L})$
 else
  $\texttt{push}(\mathcal{Q}, S, \texttt{score}(S))$

Feature Decay Algorithm

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

foreach $f \in \mathcal{F}$ **do**
    $\texttt{fvalue}(f) \leftarrow \texttt{init}(f, \mathcal{U})$
foreach $S \in \mathcal{U}$ **do**
    $\texttt{score}(S) \leftarrow \sum_{f \in \texttt{features}(S)} \texttt{fvalue}(f)$
    $\texttt{push}(\mathcal{Q}, S, \texttt{score}(S))$
while $|\mathcal{L}| < N$ **do**
    $S \leftarrow \texttt{pop}(\mathcal{Q})$
    $\texttt{score}(S) \leftarrow \sum_{f \in \texttt{features}(S)} \texttt{fvalue}(f)$
    if $\texttt{score}(S) \geq \texttt{topval}(\mathcal{Q})$ **then**
        $\mathcal{L} \leftarrow \mathcal{L} \cup \{S\}$
        foreach $f \in \texttt{features}(S)$ **do**
            $\texttt{fvalue}(f) \leftarrow \texttt{decay}(f, \mathcal{U}, \mathcal{L})$
    else
        $\texttt{push}(\mathcal{Q}, S, \texttt{score}(S))$

FDA

$$
\begin{aligned}
\texttt{init}(f,\mathcal{U}) &= 1 \text{ or } \log(|\mathcal{U}|/\texttt{cnt}(f,\mathcal{U})) \\
\texttt{decay}(f,\mathcal{U},\mathcal{L}) &= \frac{\texttt{init}(f,\mathcal{U})}{1+\texttt{cnt}(f,\mathcal{L})} \text{ or } \frac{\texttt{init}(f,\mathcal{U})}{1+2^{\texttt{cnt}(f,\mathcal{L})}}
\end{aligned}
$$

		en→de		de→en	
init	decay	*scov*	*tcov*	*scov*	*tcov*
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

Outline

1. Instance Selection for Machine Translation
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU
2. Experimental Results
   - *tcov* Comparison
   - Translation Results
   - *dice*: Instance Selection for Alignment
3. Contributions

Coverage vs. BLEU (High Coverage → High BLEU)

Effect of coverage on translation performance

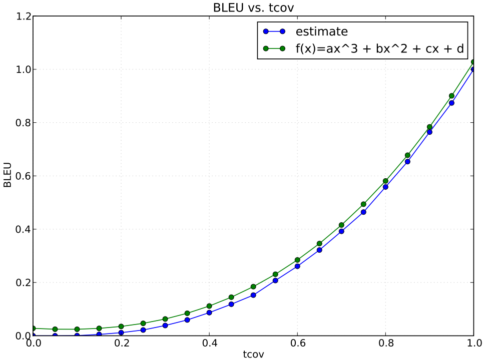

Figure: BLEU bound is a third-order function of target coverage.

$BLEU(\mathrm{T}, tcov)$

- $tcov$: percentage of target bigram features of test sentence found
- Tested: $S_i \approxeq a\ b\ \mathrm{UNK}\ d\ e$
- $BLEU(\mathbf{T}, tcov) \approx 0.56 * tcov^3 + 0.53 * tcov^2 - 0.09 * tcov + 0.003$

OUTLINE

1. INSTANCE SELECTION FOR MACHINE TRANSLATION
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU
2. EXPERIMENTAL RESULTS
   - *tcov* Comparison
   - Translation Results
   - *dice*: Instance Selection for Alignment
3. CONTRIBUTIONS

DATASET

- Train: Europarl, English-German pair: $\sim$ 1.6 million sentences.
- Dev: $26,178$ target words
- Test: $2,588$ target words
- LM: 5-gram
- *tcov*: target language 2-gram coverage

Outline

1. Instance Selection for Machine Translation
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU
2. Experimental Results
   - *tcov* Comparison
   - Translation Results
   - *dice*: Instance Selection for Alignment
3. Contributions

tcov Comparison

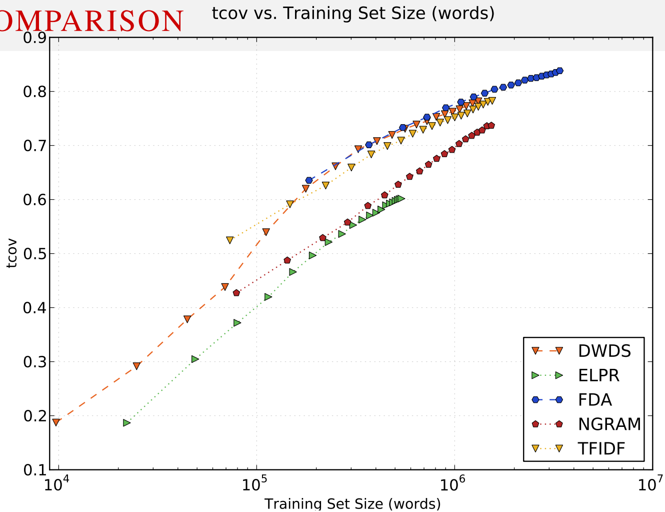

FIGURE: Target coverage curve comparison with previous work. Figure shows the rate of increase in *tcov* as the size of $\mathcal{L}$ increase.

STATISTICS OF THE OBTAINED TARGET $\mathcal{L}$

We select 1000 training instances and compare the statistics of $\mathcal{L}$.

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

Outline

1. Instance Selection for Machine Translation
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU
2. Experimental Results
   - *tcov* Comparison
   - Translation Results
   - *dice*: Instance Selection for Alignment
3. Contributions

TRANSLATION RESULTS

Moses baseline system score: 0.3577 BLEU.

We use the training instances selected by FDA in 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.

TRANSLATION RESULTS: $\mathcal{L}_{\cup}$

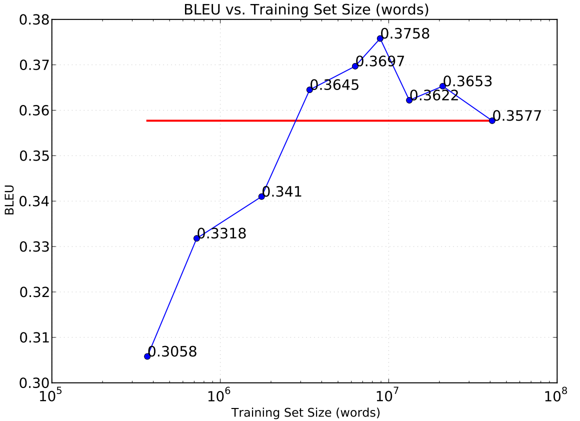

FIGURE: BLEU vs. the number of target words in $\mathcal{L}_{\cup}$.

$\Longrightarrow \sim 2$ *BLEU improvement over the baseline is possible by optimally selected subset of the training data.*

TRANSLATION RESULTS: $\mathcal{L}_{\cup\mathcal{F}}$

# 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: 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.

$\Longrightarrow$ *Within 1 BLEU performance using about* 3% *of the parallel corpus, Better performance using only about* 5%.

TRANSLATION RESULTS: $\mathcal{L}_{\mathcal{I}}$

How to obtain optimized weights?

N	100 dev sents	Mean	$\mathcal{L}_{\cup}$
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: $\mathcal{L}_{\mathcal{I}}$ performance for *en-de* using 100 sentences for tuning or mean of the weights or dev weights obtained $\mathcal{L}_{\cup}$.

$\Longrightarrow$ *Selecting the best 3000 training sentences for a specific test sentence is sufficient to obtain a score within 1 BLEU of the baseline.*

Translation Results Comparison

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

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

Outline

1. Instance Selection for Machine Translation
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU
2. Experimental Results
   - *tcov* Comparison
   - Translation Results
   - *dice*: Instance Selection for Alignment
3. Contributions

dice: Instance Selection for Alignment I

- $$dice(x, y) = \frac{2C(x, y)}{C(x)C(y)}, \tag{1}$$

  $C(x, y)$: co-occurrence count of x and y, $C(x)$: count x
- Given a test source sentence, $S_{\mathcal{U}}$, we estimate the goodness of a training sentence pair, (S, T), by the sum of the alignment scores:

$$\phi_{dice}(S_{\mathcal{U}}, S, T) = \frac{1}{|T| \log |S|} \sum_{x \in X(S_{\mathcal{U}})} \sum_{j=1}^{|T|} \sum_{y \in Y(x)} dice(y, T_j), \tag{2}$$

$X(S_{\mathcal{U}})$: features of $S_{\mathcal{U}}$, $Y(x)$: 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 $|T| \log |S|$.

dice: Instance Selection for Alignment II

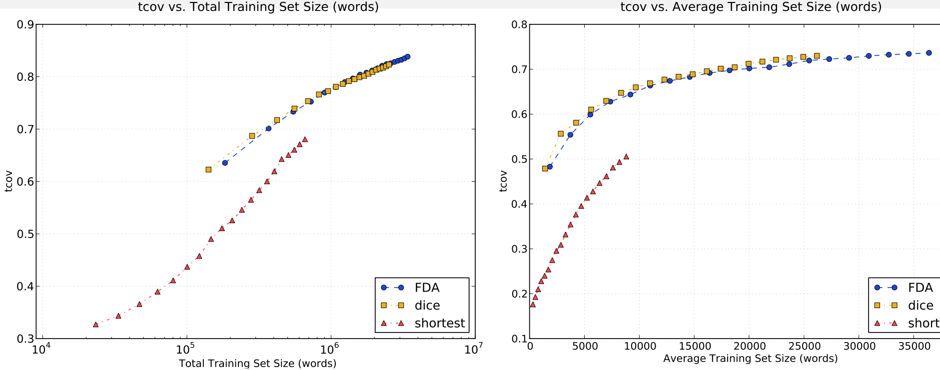

Figure: Target coverage per target words comparison. Figure shows the rate of increase in *tcov* as the size of $\mathcal{L}$ increase. Target coverage curves for total training set size is given on the left plot and for average training set size per test sentence on the right plot.

Out-of-domain Translation Results

		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: Performance for the out-of-domain translation task. ALL corresponds to the baseline system using all of the parallel corpus.

Outline

1. Instance Selection for Machine Translation
   - Related Work
   - Feature Decay Algorithm
   - High Coverage → High BLEU
2. Experimental Results
   - *tcov* Comparison
   - Translation Results
   - *dice*: Instance Selection for Alignment
3. Contributions

CONTRIBUTIONS I

- We have introduced the feature decay algorithms (FDAs), 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%.

CONTRIBUTIONS II

- We have also shown that by training on only 3000 instances per sentence we can reach within 1 BLEU difference to the baseline system.
- SMT systems can improve their performance by transductive training set selection.
- We have shown how to select instances and achieved significant performance improvements.

Thank you!

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

Eck, M., Vogel, S., and Waibel, A. (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.

Haffari, G. and Sarkar, A. (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. Association for Computational Linguistics.

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

Koehn, P. and Knight, K. (2001).
Knowledge sources for word-level translation models.
In Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing.

Lü, Y., Huang, J., and Liu, Q. (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. Association for Computational Linguistics.