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
1 Instance Selection for Machine Translation

- Related Work
- Feature Decay Algorithm
- High Coverage $\rightarrow$ High BLEU
1. **Instance Selection for Machine Translation**
   - Related Work
   - Feature Decay Algorithm
   - High Coverage $\rightarrow$ High BLEU

2. **Experimental Results**
   - $t_{cov}$ Comparison
   - Translation Results
   - $dice$: Instance Selection for Alignment
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
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
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.
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 $\rightarrow$ 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.
Instance Selection for Machine Translation

1. 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 \( \text{score} \), feature values \( \text{fvalue} \).

**Output:** Subset of the corpus to be used as the training data \( \mathcal{L} \subseteq \mathcal{U} \).

1. **foreach** \( f \in \mathcal{F} \) **do**
   
   \[ \text{fvalue}(f) \leftarrow \text{init}(f, \mathcal{U}) \]

2. **foreach** \( S \in \mathcal{U} \) **do**
   
   \[ \text{score}(S) \leftarrow \sum_{f \in \text{features}(S)} \text{fvalue}(f) \]
   
   push(\( \mathcal{Q}, \ S, \text{score}(S) \))

3. **while** \( |\mathcal{L}| < N \) **do**
   
   \[ S \leftarrow \text{pop}(\mathcal{Q}) \]
   
   \[ \text{score}(S) \leftarrow \sum_{f \in \text{features}(S)} \text{fvalue}(f) \]
   
   if \( \text{score}(S) \geq \text{topval}(\mathcal{Q}) \) **then**
   
   \[ \mathcal{L} \leftarrow \mathcal{L} \cup \{S\} \]
   
   **foreach** \( f \in \text{features}(S) \) **do**
   
   \[ \text{fvalue}(f) \leftarrow \text{decay}(f, \mathcal{U}, \mathcal{L}) \]

4. **else**
   
   push(\( \mathcal{Q}, \ S, \text{score}(S) \))
**Instance Selection for MT**

**FDA**

**Feature Decay Algorithm**

**Input**: Source corpus $\mathcal{U}$, test features $\mathcal{F}$, desired number of training instances $N$.

**Data**: Priority queue $Q$, 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** $f \in \mathcal{F}$ **do**
   - $fvalue(f) \leftarrow init(f, \mathcal{U})$

2. **foreach** $S \in \mathcal{U}$ **do**
   - $score(S) \leftarrow \sum_{f \in \text{features}(S)} fvalue(f)$
   - push($Q, S, score(S)$)

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

Ergun Biçici, Deniz Yuret (Koç University) | Instance Selection for Machine Translation | July 31, 2011 | 10 / 32
**Feature Decay Algorithm**

**Input**: Source corpus $\mathcal{U}$, test features $\mathcal{F}$, desired number of training instances $N$.

**Data**: Priority queue $Q$, sentence scores $\text{score}$, feature values $\text{fvalue}$.

**Output**: Subset of the corpus to be used as the training data $\mathcal{L} \subseteq \mathcal{U}$.

1. **foreach** $f \in \mathcal{F}$ **do**
   - $\text{fvalue}(f) \leftarrow \text{init}(f, \mathcal{U})$

2. **foreach** $S \in \mathcal{U}$ **do**
   - $\text{score}(S) \leftarrow \sum_{f \in \text{features}(S)} \text{fvalue}(f)$
   - push($Q, S, \text{score}(S)$)

3. **while** $|\mathcal{L}| < N$ **do**
   - $S \leftarrow \text{pop}(Q)$
   - $\text{score}(S) \leftarrow \sum_{f \in \text{features}(S)} \text{fvalue}(f)$
   - **if** $\text{score}(S) \geq \text{topval}(Q)$ **then**
     - $\mathcal{L} \leftarrow \mathcal{L} \cup \{S\}$
     - **foreach** $f \in \text{features}(S)$ **do**
       - $\text{fvalue}(f) \leftarrow \text{decay}(f, \mathcal{U}, \mathcal{L})$
   - **else**
     - push($Q, S, \text{score}(S)$)
**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 $\text{score}$, feature values $\text{fvalue}$.

**Output:** Subset of the corpus to be used as the training data $\mathcal{L} \subseteq \mathcal{U}$.

1. **foreach** $f \in \mathcal{F}$ **do**
   
   $\text{fvalue}(f) \leftarrow \text{init}(f, \mathcal{U})$

2. **foreach** $S \in \mathcal{U}$ **do**
   
   $\text{score}(S) \leftarrow \sum_{f \in \text{features}(S)} \text{fvalue}(f)$
   
   $\text{push}(\mathcal{Q}, S, \text{score}(S))$

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

4. **else**
   
   $\text{push}(\mathcal{Q}, S, \text{score}(S))$
FDA

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

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

<table>
<thead>
<tr>
<th>init</th>
<th>decay</th>
<th>en→de</th>
<th>de→en</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>scov</td>
<td>tcov</td>
</tr>
<tr>
<td>1</td>
<td>none</td>
<td>.761</td>
<td>.484</td>
</tr>
<tr>
<td>log(1/f)</td>
<td>none</td>
<td>.855</td>
<td>.516</td>
</tr>
<tr>
<td>1</td>
<td>1/n</td>
<td>.967</td>
<td>.575</td>
</tr>
<tr>
<td>log(1/f)</td>
<td>1/n</td>
<td>.967</td>
<td>.570</td>
</tr>
<tr>
<td>1</td>
<td>1/2^n</td>
<td>.967</td>
<td>.553</td>
</tr>
<tr>
<td>log(1/f)</td>
<td>1/2^n</td>
<td>.967</td>
<td>.557</td>
</tr>
</tbody>
</table>
Instance Selection for Machine Translation

Related Work

Feature Decay Algorithm

High Coverage → High BLEU

Experimental Results

$tcov$ Comparison

Translation Results

$dice$: Instance Selection for Alignment

Contributions
Effect of coverage on translation performance

\[ \text{BLEU}(T, tcov) \approx 0.56 \times tcov^3 + 0.53 \times tcov^2 - 0.09 \times tcov + 0.003 \]

Figure: BLEU bound is a third-order function of target coverage.
Instance Selection for Machine Translation

- Related Work
- Feature Decay Algorithm
- High Coverage $\rightarrow$ High BLEU

Experimental Results

- $t_{cov}$ Comparison
- Translation Results
- $dice$: Instance Selection for Alignment

Contributions
Results

Dataset

- Train: Europarl, English-German pair: \( \sim 1.6 \) million sentences.
- Dev: 26,178 target words
- Test: 2,588 target words
- LM: 5-gram
- \( t_{cov} \): target language 2-gram coverage
1 Instance Selection for Machine Translation
   - Related Work
   - Feature Decay Algorithm
   - High Coverage $\rightarrow$ High BLEU

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

3 Contributions
**tcov Comparison**

*tcov vs. Training Set Size (words)*

**Figure:** Target coverage curve comparison with previous work. Figure shows the rate of increase in *tcov* as the size of $\mathcal{L}$ increase.
We select 1000 training instances and compare the statistics of $\mathcal{L}$.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Unique bigrams</th>
<th>Words per sent</th>
<th>tcov</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDA</td>
<td>827,928</td>
<td>35.8</td>
<td>.74</td>
</tr>
<tr>
<td>DWDS</td>
<td>412,719</td>
<td>16.7</td>
<td>.67</td>
</tr>
<tr>
<td>TF-IDF</td>
<td>475,247</td>
<td>16.2</td>
<td>.65</td>
</tr>
<tr>
<td>NGRAM</td>
<td>626,136</td>
<td>16.6</td>
<td>.55</td>
</tr>
<tr>
<td>ELPR</td>
<td>172,703</td>
<td>10.9</td>
<td>.35</td>
</tr>
</tbody>
</table>

**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 $\rightarrow$ High BLEU

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

3 Contributions
Moses baseline system score: 0.3577 BLEU.

We use the training instances selected by FDA in three learning settings:

\[ \mathcal{L}_U: \] \( \mathcal{L} \) is the union of the instances selected for each test sentence.

\[ \mathcal{L}_{U_F}: \] \( \mathcal{L} \) is selected using all of the features found in the test set.

\[ \mathcal{L}_I: \] \( \mathcal{L} \) is the set of instances selected for each test sentence.
**FIGURE**: BLEU vs. the number of target words in $\mathcal{L}_U$.

$\Rightarrow \sim 2$ BLEU improvement over the baseline is possible by optimally selected subset of the training data.
**TRANSLATION RESULTS: \( \mathcal{L}_{\cup F} \)**

<table>
<thead>
<tr>
<th># sent</th>
<th># target words</th>
<th>BLEU</th>
<th>NIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>10,000</td>
<td>449,116</td>
<td>0.3197</td>
<td>5.7788</td>
</tr>
<tr>
<td>20,000</td>
<td>869,908</td>
<td>0.3417</td>
<td>6.0053</td>
</tr>
<tr>
<td>30,000</td>
<td>1,285,096</td>
<td>0.3492</td>
<td>6.0246</td>
</tr>
<tr>
<td>50,000</td>
<td>2,089,403</td>
<td>0.3711</td>
<td>6.1561</td>
</tr>
<tr>
<td>100,000</td>
<td>4,016,124</td>
<td>0.3648</td>
<td>6.1331</td>
</tr>
<tr>
<td><strong>ALL</strong></td>
<td><strong>41,135,754</strong></td>
<td><strong>0.3577</strong></td>
<td><strong>6.0653</strong></td>
</tr>
</tbody>
</table>

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

\[ \implies \text{Within 1 BLEU performance using about 3\% of the parallel corpus.} \]

Better performance using only about 5\%. 

Ergun Biçici, Deniz Yuret (Koç University) | Instance Selection for Machine Translation | July 31, 2011 22 / 32
Translation Results: $\mathcal{L}_I$

How to obtain optimized weights?

<table>
<thead>
<tr>
<th>N</th>
<th>100 dev sents</th>
<th>Mean</th>
<th>$\mathcal{L}_U$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.3149</td>
<td>0.3242</td>
<td>0.3354</td>
</tr>
<tr>
<td>2000</td>
<td>0.3258</td>
<td>0.3352</td>
<td>0.3395</td>
</tr>
<tr>
<td>3000</td>
<td>0.3270</td>
<td>0.3374</td>
<td>0.3501</td>
</tr>
<tr>
<td>5000</td>
<td>0.3217</td>
<td>0.3303</td>
<td>0.3458</td>
</tr>
</tbody>
</table>

Table: $\mathcal{L}_I$ performance for *en-de* using 100 sentences for tuning or mean of the weights or dev weights obtained $\mathcal{L}_U$.

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

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

3 Contributions
**dice: Instance Selection for Alignment I**

\[ dice(x, y) = \frac{2C(x, y)}{C(x)C(y)}, \]  

(1)

\( C(x, y) \): co-occurrence count of \( x \) and \( y \), \( C(x) \): count \( x \)

- Given a test source sentence, \( S_U \), we estimate the goodness of a training sentence pair, \((S, T)\), by the sum of the alignment scores:

\[ \phi_{dice}(S_U, S, T) = \frac{1}{|T| \log |S|} \sum_{x \in X(S_U)} \sum_{j=1}^{|T|} \sum_{y \in Y(x)} dice(y, T_j), \]  

(2)

\( X(S_U) \): features of \( S_U \), \( Y(x) \): 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 \(|T| \log |S|\).
**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

<table>
<thead>
<tr>
<th>BLEU</th>
<th>en-de</th>
<th>de-en</th>
<th>en-es</th>
<th>es-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>0.1376</td>
<td>0.2074</td>
<td>0.2829</td>
<td>0.2919</td>
</tr>
<tr>
<td>FDA</td>
<td>0.1363</td>
<td>0.2055</td>
<td>0.2824</td>
<td>0.2892</td>
</tr>
<tr>
<td>dice</td>
<td>0.1374</td>
<td>0.2061</td>
<td>0.2834</td>
<td>0.2857</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># target words $\times 10^6$</th>
<th>en-de</th>
<th>de-en</th>
<th>en-es</th>
<th>es-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>47.4</td>
<td>49.6</td>
<td>52.8</td>
<td>50.4</td>
</tr>
<tr>
<td>FDA</td>
<td>7.9</td>
<td>8.0</td>
<td>8.7</td>
<td>8.2</td>
</tr>
<tr>
<td>dice</td>
<td>6.9</td>
<td>7.0</td>
<td>3.9</td>
<td>3.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>% of ALL</th>
<th>en-de</th>
<th>de-en</th>
<th>en-es</th>
<th>es-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDA</td>
<td>17</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>dice</td>
<td>14</td>
<td>14</td>
<td>7.4</td>
<td>7.1</td>
</tr>
</tbody>
</table>

**Table:** Performance for the out-of-domain translation task. ALL corresponds to the baseline system using all of the parallel corpus.
Instance Selection for Machine Translation
- Related Work
- Feature Decay Algorithm
- High Coverage $\rightarrow$ High BLEU

Experimental Results
- $tcov$ Comparison
- Translation Results
- $dice$: Instance Selection for Alignment

Contributions
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%.
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!
Contributions


