Reducing Human Assessment of Machine Translation Quality to Binary Classifiers

September 8, 2007

Michael Paul, Andrew Finch, Eiichiro Sumita

NICT Spoken Language Communication Group,
ATR Spoken Language Communication Research Laboratories
Kyoto, Japan
Assessment of Machine Translation Quality

Document
- set of sentence translations
- average of sentence-level grades

Sentence
- translation of a single input
- discrete evaluation grade
- median score of multiple human grades

metrics: fluency, adequacy, ...

classifiers: SVM, DT, ...

Human
- average of sentence-level grades

Machine
- comparison to (multiple) reference translations
- assign single numerical score

metrics: BLEU, METEOR, ...

TMI 2007
©2007 NICT/ATR
Assessment of Machine Translation Quality

<table>
<thead>
<tr>
<th>Usage</th>
<th>Document</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• evaluation of MT system development progress</td>
</tr>
<tr>
<td></td>
<td>• MT system comparison (NIST, IWSLT, …)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>• quality/coverage of reference translations</td>
</tr>
<tr>
<td>• “meaning” of (numerical) automatic evaluation scores</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>• usability of given translation in a real-world application (post-editing, dialog translation, …)</td>
</tr>
</tbody>
</table>

• complexity of evaluation task (multi-class classification)
• granularity of evaluation grades
Outline of Talk

1. Prediction of Sentence-Level Translation Quality:
   - decompose multi-class to binary classification
     - a coding matrix
   - learn set of binary classifiers
     - feature selection, standard machine learning techniques
   - predict multi-class label
     - compare binary classification results to coding matrix

2. Experimental Results:
   - large-scale human-annotated evaluation corpus
   - coding matrix optimization
   - classification accuracy
   - correlation to human assessments
Prediction of Sentence-Level Translation Quality

- Training Corpus
- Feature Extraction
- Machine Learning (SVM, DT, …)
- Annotated Evaluation Corpus
- Test Set
- Feature Extraction
- Multi-class Classifier
- Binary Classifier
- Human Assessment
  - Grade = fluency
  - 5 Flawless English
  - 4 Good English
  - 3 Non-native English
  - 2 Disfluent English
  - 1 Incomprehensible
- Evaluation (classification accuracy)

(ID | Grade | F1 | F2 | …)

(ID | +1, -1)

(ID | 5, 4, 3, 2, 1)

(ID | Multi-Class)

(ID | Binary-Class)
Classification Task

Goal: predict human evaluation grade (fluency, adequacy, …) for a given translation → multi-class label

Multi-Class Classification:

😊 direct prediction of multi-class label
😔 classification accuracy is low

Binary-Class Classification:

😊 classification accuracy is high
😔 multi-class label cannot be derived reliably
Proposed Solution

Reduction of Classification Complexity:

- decompose multi-class task into a set of binary classification problems
- apply standard learning algorithm to train binary classifiers
- combine results of binary classifiers using a “coding matrix” to predict multi-class label

→ increase in classification accuracy
→ independent from learning algorithm
Proposed Solution

Feature Selection for Translation Quality Prediction:

- multiple automatic evaluation metric scores
  - BLEU
  - WER
  - GTM
  - NIST
  - PER
  - METEOR
  - TER

- metric-internal features
  - ngram-prec
  - length ratio
  - …

→ takes into account different aspects of MT quality
→ independent from target language and MT system
Prediction of Sentence-Level Translation Quality

Human Assessment

Grade = fluency
5 Flawless English
4 Good English
3 Non-native English
2 Disfluent English
1 Incomprehensible

Training Corpus

Feature Extraction

ID | Grade | F₁ | F₂ | ...

Multi-Class To Binary Class Decomposition

Machine Learning (SVM, DT, …)

ID | F₁ | F₂ | ...

Evaluation (classification accuracy)

ID | Multi-Class

+1, −1

Binary Classifier

Combination of Binary Classification Results

ID | Multi-Class
Proposed Method

1. Decomposition Phase:
   - decompose multi-class into set of binary classification tasks:
     - **one-against-all** (5, 4, 3, 2, 1):
       *Example:*
       
       5 : +1 → all training examples tagged with grade 5
       -1 → all training examples tagged with grade 4 or 3 or 2 or 1
     
     - **boundary** (54_321, 543_21):
       *Example:*
       
       54_321 : +1 → all training examples tagged with grade 5 or 4
       -1 → all training examples tagged with grade 3 or 2 or 1

     - **all-pairs** (5_4, 5_3, 5_2, 5_1, 4_3, 4_2, 4_1, 3_2, 3_1, 2_1):
       *Example:*
       
       5_4 : +1 → all training examples tagged with grade 5
       -1 → all training examples tagged with grade 4
Proposed Method

2. Learning Phase:

- **learn binary classifier** for each decomposition task
  - *feature set selection/extraction*
    - *(exp)*: + 54 features (7 autom. eval. scores + metric-internal features)
  - *classifier training*
    - *(exp)*: + fluency/adequacy, DT classifier (+ SVM classifier)
- **identify optimal subset of binary classifiers**
- **create coding matrix**
  - *column*: class of pos./neg. training examples (for given binary classifier)
  - *row*: correct binary classification result (for a given multi-class label)

3. Application Phase:

- apply all binary classifiers to given input → **classification vector** \( \mathbf{v} \)
- match \( \mathbf{v} \) against **coding matrix** rows to identify **multi-class label**
Outline of Proposed Method

Decomposition/Learning

Annotated Evaluation Corpus

Class Decomposition

Coding Matrix

Machine Learning

Binary Classifiers

Coding Matrix Optimization

Optimized Coding Matrix

Application

Source Text

MT system

Translated Text

Feature Extraction

Binary Classifiers

Binary Class Vector

Compare

Evaluation Grade
**Coding Matrix**

\[ M = \begin{pmatrix} m_{i,j} \end{pmatrix}_{i=1,\ldots,k; j=1,\ldots,l} \]

\[ m_{i,j} \in \{+1, -1, 0\} \]

(k=3, l=3)

**one-against-all**

<table>
<thead>
<tr>
<th></th>
<th>(c_1)</th>
<th>(c_2)</th>
<th>(c_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c_1)</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>(c_2)</td>
<td>-1</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>(c_3)</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
</tr>
</tbody>
</table>

**all-pairs**

<table>
<thead>
<tr>
<th></th>
<th>(c_1\bullet c_2)</th>
<th>(c_1\bullet c_3)</th>
<th>(c_2\bullet c_3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c_1)</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
</tr>
<tr>
<td>(c_2)</td>
<td>-1</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>(c_3)</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Combination of Binary Classifiers using a Coding Matrix

Hamming Distance
(number of positions that differ)

= 3
Combination of Binary Classifiers using a Coding Matrix

all-pairs

\[
\begin{array}{ccc}
  c_1 & c_2 & \text{distance} \\
  c_1 & +1 & 1 \\
  c_2 & -1 & 3 \\
  c_3 & 0 & 2 \\
\end{array}
\]

\begin{array}{ccc}
  c_1 & c_2 & c_3 \\
  c_1 & +1 & 0 \\
  c_2 & 0 & +1 \\
  c_3 & -1 & -1 \\
\end{array}
Evaluation Corpus

Basic Travel Expression Corpus (BTEC):

- 36K English translations of 4K Japanese/Chinese inputs
- human assessments and automatic evaluation scores

<table>
<thead>
<tr>
<th>sentence count</th>
<th>fluency/ adequacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>training</td>
<td>25,988</td>
</tr>
<tr>
<td>develop</td>
<td>2,024 (4 MT x 506)</td>
</tr>
<tr>
<td>test</td>
<td>7,590 (15 MT x 506)</td>
</tr>
</tbody>
</table>
Coding Matrix Optimization
(classification accuracy on DEV set)

Fluency

classification accuracy (%)
## Coding Matrix Optimization

(classification accuracy on DEV set)

### Coding Matrix

<table>
<thead>
<tr>
<th></th>
<th>54_321</th>
<th>543_21</th>
<th>5_4</th>
<th>5_3</th>
<th>5_2</th>
<th>5_1</th>
<th>4_3</th>
<th>4_2</th>
<th>4_1</th>
<th>3_2</th>
<th>3_1</th>
<th>2_1</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>+1</td>
<td>+1</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>−1</td>
<td>+1</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>−1</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
</tr>
<tr>
<td>2</td>
<td>−1</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
</tr>
<tr>
<td>1</td>
<td>−1</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
</tr>
</tbody>
</table>
Coding Matrix Optimization
(omission of worst-performing classifier)
Coding Matrix Optimization
(classification accuracy on DEV set)

Fluency

classification accuracy

%
## Coding Matrix Optimization
*(classification accuracy on DEV set)*

### Coding Matrix

<table>
<thead>
<tr>
<th></th>
<th>54_321</th>
<th>543_21</th>
<th>5_4</th>
<th>5_3</th>
<th>5_2</th>
<th>5_1</th>
<th>4_3</th>
<th>4_2</th>
<th>4_1</th>
<th>3_2</th>
<th>3_1</th>
<th>2_1</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
</tr>
<tr>
<td>4</td>
<td>+1</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
</tr>
<tr>
<td>3</td>
<td>−1</td>
<td>+1</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>−1</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
</tr>
<tr>
<td>2</td>
<td>−1</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>−1</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
</tr>
<tr>
<td>1</td>
<td>−1</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
</tr>
</tbody>
</table>

*Note: The table represents the coding matrix optimization results, with entries indicating the classification accuracy on the DEV set.*
Coding Matrix Optimization
(omission of worst-performing classifier)

Fluency

classification accuracy

omitted binary classifier

ALL 4.3

(%)

45
40
35
30
25
20
15
10
5
0
Coding Matrix Optimization
(classification accuracy on DEV set)

Fluency

classification accuracy (%)
Coding Matrix Optimization
(classification accuracy on DEV set)

<table>
<thead>
<tr>
<th></th>
<th>54_321</th>
<th>543_21</th>
<th>5_4</th>
<th>5_3</th>
<th>5_2</th>
<th>5_1</th>
<th>4_3</th>
<th>4_2</th>
<th>4_1</th>
<th>3_2</th>
<th>3_1</th>
<th>2_1</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>+1</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>−1</td>
<td>+1</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>−1</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>−1</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>−1</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
<td>−1</td>
<td>+1</td>
<td>−1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>−1</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>−1</td>
<td>0</td>
<td>0</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>−1</td>
<td>+1</td>
<td></td>
</tr>
</tbody>
</table>
Coding Matrix Optimization
(omission of worst-performing classifier)

Fluency

classification accuracy (%)

omitted binary classifier

ALL 4_3 3_2 2_1 4_2 3_1 54_321
Coding Matrix Optimization
(omission of worst-performing classifier)

Fluency

Classification accuracy

omitted binary classifier

TMI 2007
# Coding Matrix Optimization

(classification accuracy on DEV set)

## Optimized Coding Matrix

<table>
<thead>
<tr>
<th>54 _321</th>
<th>543 _21</th>
<th>5_4</th>
<th>5_3</th>
<th>5_2</th>
<th>5_1</th>
<th>4_3</th>
<th>4_2</th>
<th>4_1</th>
<th>3_2</th>
<th>3_1</th>
<th>2_1</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>
Evaluation
(classification accuracy on TEST set)

+6.0 / +6.6

49.2 / 56.0
55.2 / 62.2

classification accuracy

multi-class
coding matrix

fluency adequacy
Correlation to Human Assessment on Sentence-Level

Fluency

coding matrix  multiclass  METEOR  WER  TER  BLEU  PER  GTM  NIST

TMI 2007
©2007 NICT/ATR
Correlation to Human Assessment on Sentence-Level

Adequacy

- METEOR
- multiclass
- GTM
- WER
- PER
- NIST
- BLEU
- TER
Summary

Multiclass reduction to binary:
- **robust and reliable** method to predict human assessments on sentence-level
- **high correlation to human judges** outperforming commonly used automatic evaluation metrics
- **outperforms standard classification methods** → gains: +6.0 (fluency) and +6.6 (adequacy) in classification accuracy

Extension of proposed method:
- apply learning method to **select features used to build the coding matrix**
- investigate in the use of **additional features** that increase binary classification accuracy and thus boost overall multi-class prediction accuracy