The FBK Participation in the WMT15
Automatic Post-editing Shared Task

Rajen Chatterjee
Fondazione Bruno Kessler
chatterjee@fbk.eu

Marco Turchi
Fondazione Bruno Kessler
turchi@fbk.eu

Matteo Negri
Fondazione Bruno Kessler
negri@fbk.eu

Abstract

In this paper, we describe the “FBK English-Spanish Automatic Post-editing (APE)” systems submitted to the APE shared task at the WMT 2015. We explore the most widely used statistical APE technique (monolingual) and its most significant variant (context-aware). In this exploration, we introduce some novel task-specific dense features through which we observe improvements over the default setup of these approaches. We show these features are useful to prune the phrase table in order to remove unreliable rules and help the decoder to select useful translation options during decoding. Our primary APE system submitted at this shared task performs significantly better than the standard APE baseline.

1 Introduction

Over the last decade a lot of research has been carried out to mimic the human post-editing process in the field of Automatic Post-Editing (APE). The objective of APE is to learn how to correct machine translation (MT) errors leveraging the human post-editing feedback. The variety of data generated by human feedback, in terms of post editing, possess an unprecedented wealth of knowledge about the dynamics (practical and cognitive) of the translation process. APE leverages the potential of this knowledge to improve MT quality. The problem is appealing for several reasons. On one side, as shown by Parton et al. (2012), APE systems can improve MT output by exploiting information unavailable to the decoder, or by performing deeper text analysis that is too expensive at the decoding stage. On the other side, APE represents the only way to rectify errors present in the “black-box” scenario where the MT system is unknown or its internal decoding information is not available.

2 Statistical APE Methods

In this paper we examine the most widely used statistical phrase-based post-editing strategy proposed by Simard et al. (2007) and its most significant variant proposed by Béchara et al. (2011). We describe the two methods and their pros and cons in the following subsections.

2.1 APE-1 (Simard et al., 2007)

In this approach APE systems are trained in the same way as the statistical machine translation (SMT) system. But, as contrast to SMT which makes use of the source and target language parallel corpus, APE uses the MT output and its corresponding human post-edited data in the form of parallel corpus. One of the most important missing concepts in this “monolingual translation” is the inclusion of source information, which has been incorporated in the next approach.
2.2 APE-2 (Béchara et al., 2011)

This technique is the most significant variant of (Simard et al., 2007), where they come up with a new data representation to include the source information along with the MT output on the source side of the parallel corpus. For each MT word \( f' \), the corresponding source word (or phrase) \( e \) is identified through word alignment and used to obtain a joint representation \( f'\#e \). This results in a new intermediate language \( F'\#E \) that represents the new source side of the parallel data used to train the statistical APE system. This “context-aware” variant seems to be more precise but faces two potential problems. First, preserving the source context comes at the cost of a larger vocabulary size and, consequently, higher data sparseness that will eventually reduce the reliability of the translation rules being learned. Second, the joint representation \( f'\#e \) may be infected by the word alignment errors which may mislead the learning of translation option. Recently, Chatterjee et al. (2015) showed a fair systematic comparison of these two approaches over multiple language pairs and revealed that inclusion of source information in the form of context-aware variant is useful to improve translation quality over standard monolingual translation approach. They also showed that using monolingual translation alignment to build context-aware APE helps to mitigate the sparsity issue at the level of word alignment and for this reasons, we use this configuration to implement APE-2 method.

3 Data set and Experimental setup

Data: In this shared task we are provided with a tri-parallel corpus consisting of source (src), MT output (mt), and human post-edits (pe). While APE-1 uses only the last two elements of the triplet, all of them are used in the context-aware APE-2. To obtain joint representation \( f'\#e \) in APE-2, word alignment model is trained on src-mt parallel corpus of the training data. The training set consist of \( \sim 11K \) triplets, we divide the development set into dev and test set consisting of 500 triplets each. Our evaluation is based on the performance achieved on this test set. We tokenize the data set using the tokenizer available in the MOSES(Koehn et al., 2007) toolkit. Training and evaluation of our APE systems are performed on the true-case data.

Experiment Settings: To develop the APE systems we use the phrase-based statistical machine translation toolkit MOSES(Koehn et al., 2007). For all the experiments mentioned in this paper we use “grow-diag-final-and” as alignment heuristic and “msd-bidirectional-fe” heuristic for reordering model. MGIZA++ (Gao and Vogel, 2008) is used for word alignment. The APE systems are tuned to optimize TER(Snover et al., 2006) with MERT(Och, 2003).

We follow an incremental strategy to develop the APE systems, at each stage of the APE pipeline we find the best configuration of a component and then proceed to explore the next component. Our APE pipeline consist of various stages like language model selection, phrase table pruning, and feature designing as discussed in the following sections.

Evaluation Metric: We select TER (Snover et al., 2006) as our evaluation metric because it mimics the human post-editing effort by measuring the edit operation needed to translate the MT output into its human-revised version.

Apart from TER as an evaluation metric we also compute number of sentences being modified in the test set and then compute the precision as follow:

\[
\text{Precision} = \frac{\text{Number of Sentences Improved}}{\text{Number of Sentences Modified}}
\]

Baseline: Our baseline is the MT output as-is. To evaluate, we use the corresponding human post-edited corpus which gives us 23.10 TER score.

4 APE Pipeline

In this section we describe various components that we explore at each stage of the pipeline. At each stage, we study the effect of several configuration of each component on both the APE methods (APE-1 and APE-2)

4.1 Language Model Selection (APE-LM)

We use various data set to train multiple language models to see which of them have high impact on the translation quality. All the LMs are trained us-

---

1For each sentence in the test set, if the TER score of APE system is different than the baseline then we consider it as a modified sentence
ing IRSTLM toolkit (Federico et al., 2008) having
order of 5 gram with kneser-ney smoothing. The
data set varies in quality and quantity as described
below:

- **LM 1** contains only the training data (∼11K)
  provided in this shared task. Although the
data set contains few sentences to train a lan-
guage model compared to the data used in
MT, it is quite reliable because it is sampled
from the same distribution of the test set.

- **LM 2** consists of News Commentary having
  ∼200K sentences, downloaded from WMT
2013 translation task. This corpus belongs
to the same domain of the APE data, but it is
created under different conditions (i.e. in-
volving professional translators and translat-
ing from scratch the source sentence) making
it significantly different from the data used to
build LM1.

- **LM 3** (Big data) contains News Crawl data
  from 2007-2012 contributing to ∼13M sen-
tences, downloaded from WMT 2013 trans-
lation task. This data set has huge amount
of news crawled from the Web and covering
several topics.

- **LM1+LM2+LM3**: All the previous lan-
guage models are simultaneously used by the
APE systems. A log-linear weight is assigned
to each language model during the tuning
stage.

<table>
<thead>
<tr>
<th></th>
<th>APE-1</th>
<th>APE-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM1</td>
<td>23.95</td>
<td>24.59</td>
</tr>
<tr>
<td>LM2</td>
<td>23.96</td>
<td>24.62</td>
</tr>
<tr>
<td>LM3</td>
<td>24.06</td>
<td>24.66</td>
</tr>
<tr>
<td>LM1+LM2+LM3</td>
<td>24.05</td>
<td>24.69</td>
</tr>
</tbody>
</table>

Table 1: Performance (TER score) of the APE sys-
tems using various LMs

Results of both the APE systems are shown in
Table 1. We notice that the performance of the
APE systems do not show much variation for dif-
f erent LMs. This can come from the fact that the
news commentary and new crawl data might not
resemble well the shared task data. For this rea-
son, the in-domain LM1 is selected and used in
the next stages.

4.2 Pruning Strategy (APE-LM1-Prun)

To remove unreliable translation rules generated
from the data obtained through crowd-sourcing,
pruning strategies are investigated. First, we test
the classic pruning technique by Johnson et al.
(2007) which is based on the significance testing
of phrase pair co-occurrence in the parallel cor-
pus. According to our experiments, this technique
is too aggressive when applied on limited amounts
of sparse data. Nearly 5% of the phrase table is re-
tained after pruning with mostly self-rules (trans-
lation options that contain same source and target
phrase).

For this reason we develop a novel feature
for pruning which measures the usefulness of a
translation option present in the phrase table. For
each translation option in the phrase table, all the
parallel sentences are retrieved from the training
set such that the source phrase of the translation
option is present in the source sentence of the
parallel corpus. We then substitute the target
phrase of the translation option in the source
sentence of the parallel corpus and then compute
the TER score wrt. the corresponding target
sentence. If TER increases then we increment the
neg-count by 1, and if TER decreases we in-
crement the pos-count by 1. Finally, we com-
pute the neg-impact and the pos-impact as follows:

\[
\text{neg-impact} = \frac{\text{neg-count}}{\text{Number of Retrieved Sentences}}
\]

\[
\text{pos-impact} = \frac{\text{pos-count}}{\text{Number of Retrieved Sentences}}
\]

Once these ratios are computed for all trans-
lation options, we filter the phrase table by
thresholding on the neg-impact to remove rules
which are not useful (higher the neg-impact less
useful it is). All translation options greater than
or equal to the threshold value are filtered out.
We apply this pruning strategy for both the APE
methods over various threshold values.

Table 2 and Table 3 show the performance af-
fter pruning the APE-1-LM1 and APE-2-LM1 sys-
tems respectively. In Table 2, we observe that TER
score for various threshold values are very close
to each other, so in order to select the best thresh-
old value we base our decision on precision. So
for APE-1, we select the threshold value of 0.4
which shows the highest precision, namely APE-
1-LM1-Prun0.4. For APE-2, it is evident from
the result in Table 3 that the threshold value of 0.2
proves to be the best in terms of TER score (reduction by 1.13 point) as well as in terms of precision (APE-2-LM1-Prun0.2). These results suggest that our pruning technique has a larger impact on the APE-2 method compared to APE-1. This is motivated by the fact that the context-aware approach is affected by the data sparseness problem resulting in a large number of unreliable translation options that can be removed from the phrase table.

### 4.3 New Dense Features Design

The final stage of our APE pipeline is the feature design. When a translation system is trained using Moses, it generates translation model consisting of default dense features like phrase translation probability (direct and indirect) and lexical translation probability (direct and indirect). In the task of Automatic Post-editing where we have the source and target phrases in the same language, we can leverage this information to provide the decoder with some useful insights. In the light of this direction we design four task-specific dense features to raise the “awareness” of the decoder.

- **Similarity** ($f_1$): This feature ($f_1$) is quite similar to the one proposed in (Grundkiewicz and Junczys-Dowmunt, 2014) which measures the similarity between the source and target phrase of the translation options. The score for $f_1$ is computed as follows:

$$f_1(\text{score}) = e^{1 - \text{ter}(s, t)}$$

where $\text{ter}$ measures the number of edit operations required to translate the source phrase $s$ to the target phrase $t$ and it is computed using TER(Snoever et al., 2006).

- **Reliability** ($f_2.1$ and $f_2.2$): We allow the model to learn the reliability of the translation option by providing it with the statistics of the quality (in terms of HTER) of the parallel sentences used to learn that particular translation option. Better the quality, higher the likelihood to learn reliable rules. For each translation option in the phrase table, all the parallel sentence pairs from the training data containing the source phrase in the machine translated sentence of the pair and target phrase in the post-edited sentence are retrieved along with their HTER score. These scores are then used to compute the following two features:

  - **Median** ($f_2.1$): The median of the HTER values of all the retrieved pairs.
  - **Standard Deviation** ($f_2.2$): The standard deviation of the HTER values of all the retrieved pairs.

- **Usefulness** ($f_3$): As discussed in Section 4.2 we use $\text{pos-impact}$ as a feature to measure the positive impact of a translation option over the training set. Higher the positive impact, higher is its usefulness.

We study the impact of individual features when applied one at a time and when used all together.

<table>
<thead>
<tr>
<th>Features</th>
<th>TER</th>
<th>Number of sentences modified</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>23.87</td>
<td>81</td>
<td>0.16</td>
</tr>
<tr>
<td>$f_2.1, f_2.2$</td>
<td>23.92</td>
<td>94</td>
<td>0.19</td>
</tr>
<tr>
<td>$f_3$</td>
<td>23.88</td>
<td>82</td>
<td>0.14</td>
</tr>
<tr>
<td>$f_1, f_2.1, f_2.2, f_3$</td>
<td>23.97</td>
<td>85</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Table 4: Performance (TER score) of the APE-1-LM1-Prun0.4 for different features

Table 4 and Table 5 show the performance of various features for APE-1-LM1-prun0.4 and
Table 5: Performance (TER score) of the APE-2-LM1-Prun0.2 for different features

<table>
<thead>
<tr>
<th>Features</th>
<th>TER</th>
<th>Number of sentences modified</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1</td>
<td>23.50</td>
<td>52</td>
<td>0.27</td>
</tr>
<tr>
<td>f2.1, f2.2</td>
<td>23.50</td>
<td>53</td>
<td>0.20</td>
</tr>
<tr>
<td>f3.1</td>
<td>23.52</td>
<td>59</td>
<td>0.22</td>
</tr>
<tr>
<td>f1, f2.1, f2.2, f3.1</td>
<td>23.52</td>
<td>54</td>
<td>0.19</td>
</tr>
</tbody>
</table>

APE-2-LM1-Prun0.2 systems respectively. We observe, on this data set, that the use of these features retains the APE performance in terms of TER score but slight improvement is observed in terms of precision over both the APE systems, which indicate its contribution to improve the translation quality.

5 Final Submitted Systems

Our primary system is the best system in Table 5 i.e. APE-2-LM1-Prun0.2-f1 and contrastive system is the best system in Table 4 i.e. APE-1-LM1-Prun0.4-f2.1-f2.2. According to the shared task evaluation report the scores of our submitted systems are shown in Table 6

<table>
<thead>
<tr>
<th>Systems</th>
<th>Case Sensitive</th>
<th>Case In-sensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (MT)</td>
<td>22.91</td>
<td>22.22</td>
</tr>
<tr>
<td>APE Baseline (Simard et al., 2007)</td>
<td>23.83</td>
<td>23.13</td>
</tr>
<tr>
<td>Primary</td>
<td>23.22</td>
<td>22.55</td>
</tr>
<tr>
<td>Contrastive</td>
<td>23.64</td>
<td>22.94</td>
</tr>
</tbody>
</table>

Table 6: APE shared task evaluation score (TER)

Although we could not beat the Baseline (MT), but we see a clear improvement over APE baseline (Simard et al., 2007) by the inclusion of our novel features and the use of the pruning strategy.

6 Conclusion

The APE shared task was challenging in many terms (black-box MT, generic news domain data, crowdsourced post-editions). Though we were unable to beat the MT baseline but we gained some positive experience through this shared task. First, our primary APE system performed significantly better (0.61 TER reduction) over the standard APE baseline (Simard et al., 2007) as reported in Table 6. Second, our novel dense feature (neg-impact) used to prune phrase table shows significant improvement in the context-aware APE performance. Third, other task-specific dense features which measure similarity and reliability of the translation options help to improve the precision of our APE systems. To encourage the use of our features we have publicly released the scripts at https://bitbucket.org/turchmo/apeatfbk/src/master/papers/WMT2015/APE_2015_System_Scripts.zip.

Acknowledgements

This work has been partially supported by the EC-funded H2020 project QT21 (grant agreement no. 645452).

References


John Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phraseable.


