Why Predicting Post-Edition is so Hard? Failure Analysis of LIMSI Submission to the APE Shared Task

Guillaume Wisniewski and **Nicolas Pécheux** and **François Yvon**
Université Paris Sud and LIMSI-CNRS
91 403 ORSAY CEDEX, France
{wisniews,pecheux,yvon}@limsi.fr

Abstract

This paper describes the two systems submitted by LIMSI to the WMT'15 Shared Task on Automatic Post-Editing. The first one relies on a reformulation of the APE task as a Machine Translation task; the second implements a simple rule-based approach. Neither of these two systems manage to improve the automatic translation. We show, by carefully analyzing the failure of our systems that this counter-performance mainly results from the inconsistency in the annotations.

1 Introduction

This paper describes LIMSI submission to the WMT'15 Shared Task on Automatic Post-Editing (APE). This task aims at automatically correcting errors produced by an unknown Machine Translation (MT) system by learning from human post-editions.

For the first edition of this Shared Task we have submitted two APE systems. The first one, described in Section 3, is based on the approach of Simard et al. (2007) and considers the APE task as the automatic translation between a translation hypothesis and its post-edition. This straightforward approach does not succeed in improving translation quality. To understand the reasons of this failure, we present, in Section 4 a detailed analysis of the training data that highlights some of the difficulties of training an APE system.

The second submitted system implements a series of sieves, applying, each, a simple post-editing rule. The definition of these rules is based on our analysis of the most frequent error corrections. Experiments with this approach (Section 5) show that this system also hurts translation quality. However, analyzing its failures allows us to show that the main difficulties in correcting MT errors result from the inconsistency between the different post-editions.

2 Data Preprocessing

The Shared Task organizers provide training and development data that consist of respectively 11,272 and 1,000 examples. Each example is made of an English source sentence, its automatic translation in Spanish by an unknown MT system and a human revision of this translation. All sentences are tokenized. There are, on average, 22.88 words in each post-edition, the longest post-edition having 199 words and the shortest 3.

In a first pre-processing step we have removed all examples for which the ratio between the length of the automatic translation and the length of the corresponding post-edition was higher than 1.2 or lower than 0.8. As shown in Table 1, these examples correspond mainly to errors in sentence boundaries or to 'over-translation' (e.g. when the post-editor added the translated title in the third example of Table 1), that could have a negative impact on the training of an APE system. At the end, the training set we used in all our experiments is made of 10,404 sentences.

The source sentences and the automatic translation of the training and development set have been aligned at the word level using FASTALIGN (Dyer et al., 2013) and the grow-diag-final symmetrization heuristic. To improve alignment quality, the sources and the translations have been first concatenated to the English-Spanish Europarl dataset and the resulting corpus has been aligned as a whole. Spanish MT outputs and post-editions have also been PoS-tagged using FREELING,[1] a state-of-the-art rule-based PoS tagger for Spanish. We used a CRF-based model trained on the Penn Treebank for the English source sentences. All PoS tags have been mapped to the universal PoS

[1] http://nlp.lsi.upc.edu/freeling/

Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 222–227,
Lisboa, Portugal, 17-18 September 2015.

src n°3334	Gomez Flies To Miami To Be With Bieber !
tgt n°3334	Gómez Vuela a Miami para estar con Bieber !
pe n°3334	Gómez Vuela hasta Miami para estar con Bieber ! AQUÍ estan las Pruebas ! Parece que estos dos tortolitos están juntos de nuevo y esta vez , podrian estar cantando .. La pelea de Twitter entre Demi Lovato y Kathy Griffin fue tan serio que hasta se involucro la policia y hubieron amenazas de muerte !
src n°517	that are sooooo good !
tgt n°517	que son taaaan bueno !
pe n°517	La favorita de Perezcious , Lissie , acaba de lanzar un nuevo EP de covers ... ¡ que están taaaan buenos !
src n°4444	MAJOR Amazing Spider-Man 2 Spoiler Alert !
tgt n°4444	MAJOR Amazing Spider-Man 2 Spoiler Alert !
pe n°4444	GRAN Alerta de Spoiler para The Amazing Spider-Man 2 (El maravilloso Hombre Araña 2) !

Table 1: Examples of automatic translations and their post-editions for which the ratio between their length is higher than 1.2.

tagset of Petrov et al. (2012) to make interpretation easier. Note that these two procedures are error-prone (especially as we have no information about the tokenization) and may introduce some noise in our analysis (cf. Section 4).

We have also computed an edit distance between the automatic translations and their post-editions using Python standard `difflib` module that allows us to define an 'alignment', at the phrase-level,[2] between these two sentences. The `difflib` module implements the Ratcliff-Obershelp algorithm (Ratcliff and Metzener, 1988) that finds a sequence of edits transforming a sentence into another. While this sequence is not necessarily of minimal length, it is faster to compute, easier to use and, above all, more interpretable than the one computed using the standard minimum edit distance algorithm. In particular, `difflib` is able to automatically find edits between 'phrases' rather than between single words.

3 Automatic Post-Editing as Machine Translation

The first system we have developed for the Shared Task is inspired by the approach of Simard et al. (2007) and reduces the Automatic Post-Edition task as a Machine Translation task. Ignoring the source sentence, we train a standard phrase-based machine translation system using the automatic translation as a source sentence and its post-edition as the target sentence.

The word alignment between the automatic translation and the post-edited sentence, used as input in our APE-MT pipeline, has been computed using Meteor (Denkowski and Lavie, 2014). The APE-MT system has then been trained following the usual steps.[3] In our experiments, we used our in-house MT system NCODE (Crego et al., 2011) that implements a n-gram based translation model. As main features we used a 3-gram bilingual language model on words, a 4-gram bilingual language model on PoS factors and a 4-gram target language model trained only on the post-editions sentences, along with the conventional features (4 lexical features, 6 lexicalized reordering, distortion model, word and phrase penalty). We did allow reorderings during decoding. The training data is used to extract and compute the different models while the development data is used to perform the tuning step.

The results, evaluated by the hTER score[4] between the predicted and the human post-editions, are summarized in Table 2. This straightforward approach actually hurts performance and the results show that we are not able to predict post-editions: the output of the MT system is closer to the post-edition than the prediction of our APE-

[2] As usual in MT, we use 'phrase' to denote a sequence of consecutive words.

[3] see, for instance, `https://ncode.limsi.fr`

[4] All reported hTER scores are case-sensitive and have been computed using the scripts provided by the Shared Task organizers.

	train	development	test
MT output	23.32	23.21	22.91
APE-MT output	21.64	23.95	23.57

Table 2: hTER score achieved by MT system train to predict the post-edition from the MT output.

MT system. This is true even for the development data on which our system was tuned.

4 Data Analysis

To understand the results of our first APE model, we analyzed thoughtfully the data provided by the shared task organizers.

The risk of over-correcting The first important observation is that the MT system used to translate the source sentences achieves an hTER score of 23.32 on the training data, meaning that, roughly, more than three words out of four are correct and must not be modified. As a consequence, predicting which words must be post-edited is an highly unbalanced problem. It is, therefore, very likely that any modifications of the MT output could hurt translation quality. Let n denote the number of word of in the dataset and a the percentage of words that are mistranslated. If we are able to detect mistranslated words with a precision p and a recall r and to correct them with precision c, the number of errors after the automatic post-editing equals to the sum of the number of errors that have not been corrected $(n \times a \times (1 - r))$, the number of errors the correction of which is erroneous $(n \times a \times r \times (1 - c))$ and of the number correct words that have been modified $(n \times a \times r \times (1 - p) \div p)$. For the shared task training data, $n = 238,332$, $a = 0.25$ and we assume that $c = 0.8$, which is an optimistic estimate. To avoid introducing new error, the F_1 score of the system detecting mistranslated word must be higher that 0.7, which is far better than the performance achieved by most state-of-the-art word-level confidence estimation system.

Uniqueness of edits To characterize annotators edits, we have computed the distribution of the three basic operations (Table 3) as well as the 20 most frequent 'lexicalized' edits (Table 4). Several observations, similar to the findings of our analysis of an English-French post-editions corpus (Wisniewski et al., 2013), can be made from these two tables. First, and most importantly, it appears that most edits are unique: even the most frequent edit (insertion of '¡') only accounts for a negligible part of all edits. Overall, 24.74% of all edits are unique. As a consequence, it is very unlikely that any approach, such as the one described in Section 3, that relies solely on word-level pattern recognition and transformation, will be able to generalize the observed corrections to new sentences. This explains why our APT-MT systems improves on the training data, on which transformation where learned, but fails to generalize (Table 2).

operation	*count*	%
deletion	$4,795$	15.56%
insertion	$5,873$	19.07%
substitution	$20,129$	65.37%
total	$30,797$	100%

Table 3: Distribution of the edit types in the training set.

edits	occurrences	edits	occurrences
+¡	286	+la	108
+,	267	-el	107
+de	247	+el	102
+que	231	-los	101
-,	202	+los	92
-que	164	-se	92
-la	164	+en	88
+a	156	+se	85
-de	146	su → tu	71
+'	117	+las	68

Table 4: Most frequent post-edits on the training set. Additions and deletion are denoted by '+' and '-'; substitutions by '→' .

Importance of edits related to punctuation Second, it appears that the most frequent edits are mainly insertions or deletions of either a frequent word or a punctuation. Table 5 shows the distribution of edits that concern *only* punctuations. These edits account for an important part of all the modifications made by the post-editors: correcting them automatically would reduce the hTER score by more than 3 points. Some of these edits correspond to genuine translation errors that must be corrected for the output sentence to be gram-

edits	count	%
addition	581	1.88
deletion	394	1.27
substitution	85	0.27
Total	1,060	3.42

Table 5: Number of edits involving *only* punctuation.

Accesorios → accesorios	Guía → guía
Campo → campo	está loco → Está Loco
algas → alGAS	Inglés → inglés
legión → Legión	poderes → PODERES
thefamily → TheFamily	mucho → MUCHO

Table 6: Examples of substitutions that involve only changes in case.

substitution	count
VERB → VERB	2,372
NOUN → NOUN	1,243
ADP → ADP	605
ADJ → ADJ	571
PRON VERB → VERB	225
DET → DET	224
VERB → NOUN	178
NOUN → VERB	169
DET NOUN → DET NOUN	151
NOUN → ADJ	147
NOUN → DET NOUN	146
ADV → ADV	136
DET NOUN → NOUN	119
PRON → PRON	109
ADJ → NOUN	89
VERB ADP → VERB	76
total	6,560

Table 7: PoS of the words involved in a substitution.

matically correct. In particular, in Spanish, all interrogative and exclamatory sentences or clauses have to begin with an inverted question mark (¿) or exclamation mark (¡). These long-range dependencies are difficult to capture with a phrase-based system, which explains why inverted punctuation often have to be inserted by the post-editors. However, many other modifications (especially the insertion and deletion of comas) are more an improvement of style and their presence in a 'minimal' post-edition can be questioned.

We will now consider the most frequent types of edits and focus on three different kind of substitutions.

Importance of edits related to change in case We first looked at changes in case: it appears that 1.16% of all edits are solely a change in case. Table 6 gives some examples of such edits. The high proportion of edits related to case is not really surprising as it can be assumed that the MT system has been trained on lower-cased data and its output has been re-cased in a second, independent step, which is a difficult task. However, as for the punctuations, word case rarely affects the meaning of a sentence and its correction can be considered more as 'normalization' rather than 'mandatory' edits.

Correcting verb endings To better characterize the different kind of substitutions, we have represented, Table 7, the PoS of the words involved in a substitution. This table shows that many of the substitutions that occur during post-edition keep the grammatical structure of the sentence unchanged and only modify lexical choices: in 26.7% of the substitutions, the PoS of the words that are edited are kept unchanged. Interestingly, as for lexicalized edits presented in Table 4 most of the 'PoS substitutions' are unique. But, when looking at the tail of the distribution, it appears that many of these unique transformations are due to error in alignment (e.g. when a single word is replaced by 6 or 7 words) or to error in PoS prediction.

Looking more closely at verb modifications, it appears that, in 39.68% of them, the prefix[5] of the words is the same, suggesting that a lot of edits consist in changing the verb conjugation, which might be surprising as it could be expected that the language model would resolve such difficulties. Table 8 gives some examples of verb post-editings. Surprisingly, this observation is no longer true for modifications of nouns: in less than 10% of them, the prefix is the same before and after post-editing.

5 A Multi-Sieve Approach to Automatic Post-Editing

5.1 Main Principles

We consider a simple Automatic Post-Edition architecture based on a sieve that applies simple post-editing rules. Using such a simple rule-based approach has two main motivations. First, by focusing on very precise categories of errors, we expect to avoid 'over-correcting' the translation hypotheses as our APE-MT model; second, analyz-

[5]The prefix is defined as the first five letters of a word.

same prefix	different prefix
piensa → piense (thinks)	significa → representa (means)
escritos → escritas (NULL)	significa → representa (NULL)
guardar → guardan (save)	superar → batir beat
afeitado → afeitadas (shaven)	preocupa → ocupa preoccupies
visita → visitas (visit)	Ofender → ofendiendo Offending
tratando → tratar (trying)	metió → metí (NULL)
adecuado → adecuada (suited)	tengo → conseguí (I)
presentan → presente (come)	dejar → deje (quit)
pregunta → preguntaste (asking)	seguir → cumplir (keep)
enseñado → enseñó (taught)	invertido → investido (invested)

Table 8: Example of verb substitutions with the source word they are aligned with.

ing the errors of these simple rules will be much easier than analyzing the output of a complete MT system such as the one presented in Section 3 and we expect to gain some insights about the interplay between the different factors at stake.

In this work, we have considered three post-editing rules that correspond to the main categories of errors identified in Section 4. These rules aim at:

- predicting word case;
- predicting exclamation and interrogation marks;
- predicting verbal endings.

Prediction of word case We used a very naive approach to predict the case of a word by assuming that a translated word should have the same case as the source word it is aligned with. We therefore converted all words that were aligned with a lower-cased, upper-cased or title-cased word to their lower-cased, upper-cased or title-cased version, respectively. To account for missing alignment links, we also converted all target word in upper-case when all the words of the source sentence were upper-cased.

Prediction of exclamation and interrogation marks As explained in Section 4, in Spanish, interrogative and exclamatory sentences or clauses have to begin with an inverted question mark (¿) or exclamation mark (¡). We use the method described in Algorithm 1 to insert question marks[6] at the beginning and end of clauses. This method simply inserts the same punctuation mark as in the source sentence[7] at the end of the sentence and finds the beginning of the clause by looking for a set of specific characters to insert the inverted punctuation mark right after it. When the beginning of the clause can not be found, the inverted punctuation mark is inserted at the beginning of the sentence.

Algorithm 1: Insert question marks at end and beginning of clauses .

input: $s = (s_i)_{i=1}^{|s|}$ a source sentence
remove '?' and '¿' from target sentence
if *'?'* $\notin s$ **then**
  return s
add '?' at end of target sentence
for $i \in [\![|s|, 0]\!]$ **do**
  if $i = 0$ or $s_i \in$ *'–:,"-'* **then**
    insert '¿' at the (i+1)$^{\text{th}}$ position
    break

Correcting Verbs Ending We used a two-step models to correct verb endings. In a first step we generate, for each verb identified in the translation hypothesis, a list of candidates containing conjugation variants for this verb form. We then choose the verb form which maximizes the language model score of the modified sentence as the correction. To generate the list of candidates, we extracted automatically the conjugation tables of Spanish Wiktionary[8], building a list of 587,832 verb forms with their lemma. We used, as a scoring model, a 5-gram language model trained on the Spanish data of the WMT campaign.

This post-edition rule is more prone to errors than the previous two rules as it relies on a language model (that was trained on data with a different tokenization) and on an external resource to generate the candidates (that is neither complete nor completely accurate).

5.2 Experimental Results

Table 9 shows the result, evaluated on the Shared Task development set, of the multi-sieve approach described in the previous section. As for the MT model presented in Section 3, our model degrades translation quality, even if it makes only a small number of precise modifications, showing that there are more errors introduced by our multi-

[6]The same method was used to insert exclamation marks.

[7]Only inserting the inverted punctuation mark slightly hurts performance: it appears that not all interrogative sentence are translated into an interrogative sentence.

[8]es.wiktionary.org

	hTER
baseline	23.320
+case correction	23.396
+punctuation correction	23.708
+verb correction	24.217

Table 9: hTER score achieved by our multi-sieve approach on the development data.

sieve approach than there are errors that are corrected.

The analysis of our errors shows that the observed drop in performance can be explained by the inconsistencies in the post-editions. For instance, in the case of interrogative sentences, there are 558 translation hypotheses in the training set that end with an interrogative mark, 203 of which do not contain an inverted mark. Applying Algorithm 1, will correct all of them. However, it also appears that, in 108 of these 203 sentences (53%) no inverted interrogative marks were added by the post-editors — resulting in 'un-grammatical' sentences. At the end, even the correct introduction of inverted question marks would make translation hypotheses less similar to the human post-edition. A similar observation can be made for the exclamatory sentences.

Regarding the correction of case, the proposed post-edition rule achieves very good performance when its application is restricted to the word that have to be post-edited (i.e. when using the post-edition as an oracle to identify which words must be corrected): it is able to correctly predict the case of the word in almost 85% of the case. The erroneous corrections mainly result from alignment errors. However, when applied on the whole corpora it will also change the case of many words the post-editors have not modified. When we looked at these words we did not see any reasons why they should not have been modified.

6 Discussion and Conclusion

We described two different approaches to Automatic Post-Editing: the first one casts the problem as a monolingual MT task; the second one uses a series of simple, yet effective, post-edition sieves. Unfortunately, none of our systems was able to outperform the simplest do-nothing baseline. While better post-editions methods have yet to be found, we argue that this negative result is mainly explained by the difficulty of the task at hand and the small amount of available data. Indeed, none of the participants to this pilot Shared Task managed to outperform the baseline. This is confirmed by an in-depth analysis of the task which shows that: (a) most of the post-edition operations are nearly unique, which makes very difficult to generalize from a small amount of data; and (b) even when they are not, inconsistencies in the annotations between the different post-editions prevent from improving over the baseline.

Acknowledgments

This work was partly supported by the French "National Research Agency" (ANR) under project ANR-12-CORD-0015/Transread.

References

Josep Maria Crego, François Yvon, and José B. Mariño. 2011. N-code: an open-source Bilingual N-gram SMT Toolkit. *Prague Bulletin of Mathematical Linguistics*, 96:49–58.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *NAACL*, pages 644–648, Atlanta, Georgia, June. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *LREC*, Istanbul, Turkey, may.

John W. Ratcliff and D. E. Metzener. 1988. Pattern matching: The gestalt approach. *Dr. Dobb's Journal*.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 508–515, Rochester, New York, April. Association for Computational Linguistics.

Guillaume Wisniewski, Anil Kumar Singh, Natalia Segal, and François Yvon. 2013. Design and analysis of a large corpus of post-edited translations: quality estimation, failure analysis and the variability of post-edition. *Machine Translation Summit*, 14:117–124.