

Can the Internet help improve Machine Translation?

Ariadna Font Llitjós

Language Technologies Institute

Carnegie Mellon University

Pittsburgh, PA, 15213. USA

aria@cs.cmu.edu

Abstract

This paper summarizes a largely automated method that uses online post-editing feedback to automatically improve translation rules. As a starting point, bilingual speakers' local fixes are collected through an online Translation Correction Tool. Next, the Rule Refinement Module attacks the problem at its core and uses the local fixes to detect incorrect rules that need to be refined. Once the grammar and lexicon have been refined, the Machine Translation system not only produces the correct translation as fixed by the bilingual speaker, but is also able to generalize and correctly translates similar sentences. Thus, this work constitutes a novel approach to improving translation quality. Enhanced by the reaching power of the Internet, our approach becomes even more relevant to address the problem of how to automatically improve the quality of Machine Translation output.

1 Introduction

Achieving high translation quality remains the biggest challenge Machine Translation (MT) systems currently face. Researchers have explored a variety of methods to include user feedback in the MT loop. Similar to our approach, Phaholphyinyo and colleagues (2005) proposed adding post-editing rules to their English-Thai MT system with the use of a post-editing tool. However, they use context sensitive pattern-matching rules, which make it impossible to fix errors involving missing words.

Unlike our approach, in their system, the rules are created by experienced linguists and their approach requires a large corpus. They mention an experiment with 6,000 bilingual sentences but report no results due to data sparseness.

In general, most MT systems have failed to incorporate post-editing efforts beyond the addition of corrected translations to the parallel training data for SMT and EBMT or to a translation memory database.¹ Therefore, a largely automated method that uses online post-editing information to automatically improve translation rules constitutes a great advance in the field.

If an MT-produced translation is incorrect, a bilingual speaker can diagnose the presence of an error reliably using the online Translation Correction Tool (Font Llitjós and Carbonell, 2004). An example of an English-Spanish sentence pair generated by our MT system is “*Gaudí was a great artist - Gaudí era un artista grande*”. Using the online tool, bilingual speakers modified the incorrect translation to obtain a correct one: “*Gaudí era un gran artista*”.

Bilingual speakers, however, cannot be expected to diagnose which complex translation rules produced the error, and even less, determine how to improve those rules. One of the main goals of this research is to automate the Rule Refinement process based on just *error-locus* and possibly some *error-type* information from the bilingual speaker, relying on rule blame assignment and on regression testing to evaluate and measure the consequent improvement in MT accuracy. In this case, our Automatic Rule Refinement system can add the missing sense to the lexicon (*great*→*gran*) as

¹ For a more detailed discussion, see Font Llitjós and colleagues (2005a)

well as the special case rule for Spanish pre-nominal adjectives to the grammar.

With this system in place, we envision a modified version of the Translation Correction Tool as a game with a purpose, available online through a major web portal. This would allow bilingual speakers to correct MT input and get rewards for making good corrections, and compare their scores and speed with other users. For the MT community this means having a free and easy way to get MT output feedback and potentially improve their systems based on such feedback. Furthermore, a fully interactive system would be a great opportunity to show users that their corrections have a visible impact on technology, since they would see the effects their corrections have on other sentences. Last but not least, this new method is also expected to be particularly useful in resource-poor scenarios, such as the ones the Avenue project is devoted to (Font Llitjós et al., 2005b), where statistical systems are not an option and where there might be no experts with knowledge of the resource-poor language (Figure 1).

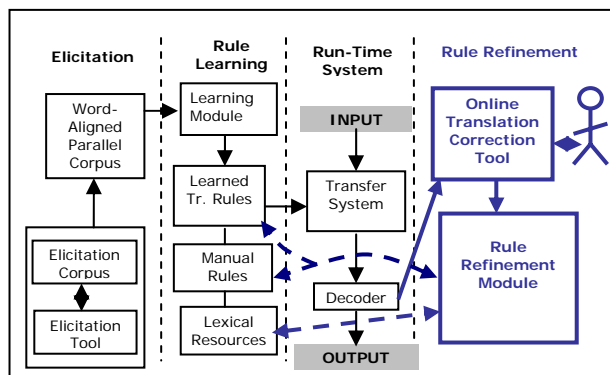


Figure 1. Simplified Avenue Architecture

2 Online Elicitation of MT Errors

The main challenge of the error elicitation part of this work is how to elicit minimal post-editing information from non-expert bilingual speakers. The Translation Correction Tool (TCTool) is a user-friendly online tool that allows users to add, delete and modify words and alignments, as well as to drag words around to change word order. A set of user studies was conducted to discover the right amount of error information that bilingual speakers can detect reliably when using the TCTool. These studies showed that simple error information can be elicited much more reliably (F1 0.89) than error

type information (F1 0.72) (Font Llitjós and Carbonell, 2004). Most importantly, it became apparent that for our Rule Refinement purposes, the list of correction action(s) with information about error and correction words is sufficient.

Building on the example introduced above, Figure 2 shows the initial state of the TCTool, once the user has decided that the translation produced by the MT system is not correct.

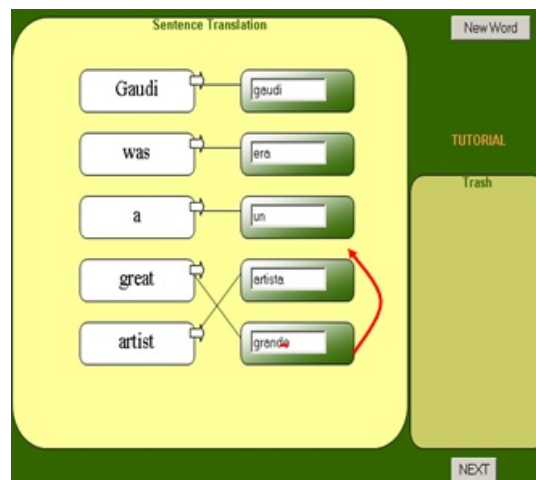


Figure 2. TCTool snapshot with initial translation pair

In this case, the bilingual speaker changed ‘grande’ to ‘gran’ and dragged ‘gran(de)’ in front of ‘artista’, effectively flipping the order of these two words. Figure 3 shows the state of the TCTool after the user corrections.

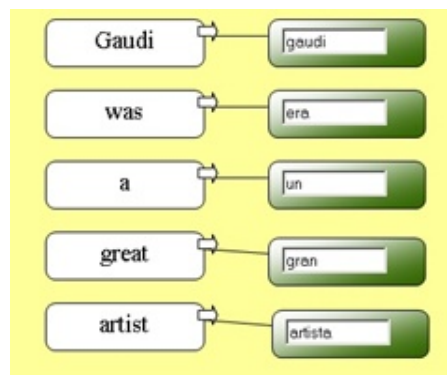


Figure 3. TCTool snapshot after user has corrected the translation

3 Extracting Error Information

User correction actions are registered into a log file. The Automatic Rule Refinement (RR) module extracts all the relevant information from the

TCTool log files and stores it into a Correction Instance. See Figure 4 for an example.

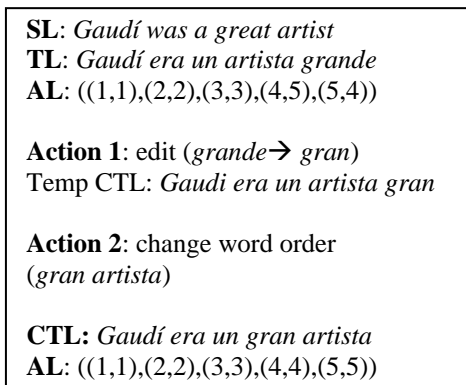


Figure 4. A Correction Instance stores the source language sentence (SL), the target language sentence (TL) and the initial alignments (AL), as well as all the correction actions done by the user. It also provides the corrected translation (CTL) and final alignments.

The Rule Refinement (RR) module processes one action at a time. So in this approach, the order in which users correct a sentence does have an impact on the order in which refinements apply.

4 Lexical Refinements

After having stored all the relevant information from the log file, the Rule Refinement module starts processing the Correction Instance. In the example above, it first goes into the lexicon and, after double checking that there is no lexical entry for [great → gran], it proceeds to add one by duplicating the lexical entry for [great → grande]. Since these two lexical entries are identical at the feature level, the RR module postulates a new binary feature, say *feat1*², which serves the purpose of distinguishing between two words that are otherwise identical (according to our lexicon):

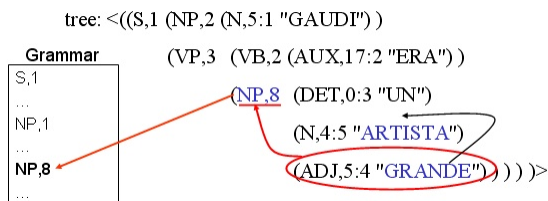
```
ADJ::ADJ | : [great] -> [grande]
((X1::Y1)
((x0 form) = great)
((y0 agr num) = sg)
((y0 agr gen) = masc)
((y0 feat1) = -))

ADJ::ADJ | : [great] -> [gran]
((X1::Y1)
((x0 form) = great)
((y0 agr num) = sg)
((y0 agr gen) = masc)
((y0 feat1) = +))
```

² A more mnemonic name for *feat1* would be *pre-nominal*.

5 Rule Refinements

Now the RR module moves on to process the next action in the Correction Instance and the first step is to look at the parse trace output by the MT system, so that the grammar rule responsible for the error can be identified:



At this point, the system extracts the relevant rule (NP,8) from the grammar, and has two options, either to make the required changes directly onto the original rule (REFINE) or to make a copy of the original rule and modify the copy (BIFURCATE). If the system has correctly applied the rule in the past (perhaps because users have evaluated the translation pair “*She saw a dangerous man – Ella vio un hombre peligroso*” as correct), then the RR module opts for the BIFURCATE operation. In this case, the RR module makes a copy of the original rule (NP,8) and then modifies the copy (NP,8’) by flipping the order of the noun and adjective constituents, as indicated by the user. This rule needs to unify with ‘gran’ but not with ‘grande’, and so the RR module proceeds to add the constraint that the Spanish adjective (now *y2*) needs to have the *feat1* with value +:

```
{NP,8}
NP::NP : [DET ADJ N] -> [DET ADJ N]
( (X1::Y1) (X2::Y2) (X3::Y3)
((x0 def) = (x1 def))
(x0 = x3)
((y1 agr) = (y3 agr)); DET-N agreement
((y2 agr) = (y3 agr)); ADJ-N agreement
(y2 = x3)
((y2 feat1) = c + ))
```

These two refinements result in the MT system generating the desired translation, namely “*Gaudí era un gran artista*” and not the previous incorrect translation. But can the system also eliminate other incorrect translations automatically? In addition to generating the correct translation, we would also like the RR module to produce a refined grammar that is as tight as possible, given the data that is available. Since the system already has the information that “*un artista gran*” is not a correct se-

quence in Spanish, the grammar can be further refined to also rule out this incorrect translation. This can be done by restricting the application of the general rule (NP,8) to just post-nominal adjectives, like ‘grande’, which in this example are marked in the lexicon with (*feat1* = -).

6 Generalization power

The difference between this approach and mere post-editing is that the resulting refinements affect not only to the translation instance corrected by the user, but also to other similar sentences where the same error would manifest. After the refinements have been applied to the grammar in our example sentence, a sentence like “*Irina is a great friend*” will now correctly be translated as “*Irina es una gran amiga*”, instead of “*Irina es una amiga grande*”.

7 Evaluation

We plan to evaluate the RR module on its ability to improve coverage and overall translation quality. This requires identifying sensible evaluation metrics. Initial experiments have shown that both BLEU [Papineni et al., 2001] and METEOR [Lavie et al., 2004] can automatically distinguish between raw MT output and corrected MT output, even for a small set of sentences. In addition to the presence of the corrected translation in the lattice produced by the refined system, our evaluation metrics will also need to take into account whether the incorrect translation is now prevented from being generated and whether the lattice of alternative translations increased or decreased. A decrease of lattice size would mean that the refinement also made the grammar tighter, which is the desired effect.

8 Technical Challenges and Future Work

The Rule Refinement process is not invariable. It depends on the order in which refinement operations are applied. In batch mode, the RR module can rank Correction Instances (CI) in such a way as to maximize translation accuracy. Suppose that the first CI (CI1) triggers a bifurcation of a grammar rule, like the one we see in the example described in Section 5. After that, any CI that affects the same rule that got bifurcated, will only modify the original rule (NP,8) and not the copy (NP,8’).

If the constraint that enforces determiner-noun agreement were missing from the original rule, say, the copy (NP,8’) would not have that constraint added to it, and so another example with the pre-nominal adjective exhibiting that agreement error would be required (CI2: **Irina es un gran amiga*), before the system added the relevant constraint to NP,8’. However, if we can detect such rule dependencies before the refinement process, then we can try to find an optimal ranking, given the current set of CIs, which should result in higher translation accuracy, as measured on a test set.

Another interesting future direction is enhancing the Rule Refinement system to allow for further user interaction. In an interactive mode, the system can use Active Learning to produce minimal pairs to further investigate which refinement operations are more robust, treating the bilingual speaker as an oracle. We hope to explore the space between batch mode and a fully interactive system to discover the optimal setting which allows the system to only ask the user for further interaction when it cannot determine the appropriate refinement operation or when it would be impossible to correctly refine the grammar and the lexicon automatically.

References

- Alon Lavie, Kenji Sagae and Shyamsundar Jayaraman. 2004. *The Significance of Recall in Automatic Metrics for MT Evaluation*. AMTA, Washington, DC.
- Ariadna Font Llitjós, Jaime Carbonell and Alon Lavie. 2005a. *A Framework for Interactive and Automatic Refinement of Transfer-based Machine Translation*. EAMT, Budapest, Hungary.
- Ariadna Font Llitjós, Roberto Aranovich and Lori Levin. 2005b. *Building Machine translation systems for indigenous languages*. Second Conference on the Indigenous Languages of Latin America (CILLA II), Texas, USA.
- Ariadna Font Llitjós and Jaime Carbonell. 2004. *The Translation Correction Tool: English-Spanish user studies*. LREC 04, Lisbon, Portugal.
- Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2001. *BLEU: A Method for Automatic Evaluation of Machine Translation*. IBM Research Report RC22176 (W0109-022).
- Siththa Phaholphinyo, Teerapong Modhiran, Nattapol Kritsuthikul and Thepchai Supnithi. 2005. *A Practical of Memory-based Approach for Improving Accuracy of MT*. MT Summit X. Phuket Island, Thailand.