The Translation Correction Tool: English-Spanish user studies

Ariadna Font Llitjós and Jaime Carbonell

Language Technologies Institute
Carnegie Mellon University,
Pittsburgh, PA 15213, USA
{aria, jgc}@cs.cmu.edu

Abstract

Machine translation systems should improve with feedback from post-editors, but none do beyond the very localized benefit of adding the corrected translation to parallel training data (for statistical and example-base MTS) or a memory data base. Rule based systems to date improve only via manual debugging. In contrast, we introduce a largely automated method for capturing more information from the human post-editor, so that corrections may be performed automatically to translation grammar rules and lexical entries. This paper focuses on the information capture phase and reports on an experiment with English-Spanish translation.

1. Introduction

Whereas machine translation (MT) has been developed for many language pairs to reduce the cost of human translation, MT has not yet demonstrated human-quality translation, and requires significant post-editing. The work of post-editing, however, is seldom recycled into MT system improvements, and never in a fully automated way. The objective of our research is to complete that essential feedback loop in as automated a manner as possible, requiring some extra work by the post-editor, but not requiring any specialized linguistic training or programming skill.

2. AVENUE project: learning and refining translation rules automatically

Our MT research group at Carnegie Mellon has been working on a new MT approach, under the AVENUE project, that is specifically designed to enable rapid development of MT for languages with limited amounts of online resources.

Our approach assumes the availability of a small number of bilingual speakers of the two languages, but these need not be linguistic experts. The bilingual speakers create a comparatively small corpus of word aligned phrases and sentences (on the order of magnitude of a few thousand sentence pairs) using a specially designed elicitation tool (Probst et al., 2001).

From this data, the learning module of our system automatically infers hierarchical syntactic transfer rules, which encode how constituent structures in the source language (sl) transfer to the target language (tl). The collection of transfer rules, which constitute the translation grammar, is then used in our run-time system to translate previously unseen sl text into the tl (Probst et al., 2003).

Our transfer-based MTS consists of four main modules: elicitation of a word aligned parallel corpus; automatic learning of translation rules; the run time transfer system, and the interactive and automatic refinement of the translation rules.

The TCTool is the first necessary step for the last module in our MTS. It is the front-end that interacts with users to extract their judgements and corrections of the current translations produced by our MTS. User feedback is the input to the second part of the last module: the automatic refinement of translation rules.

3. MT error classification

There are several ways to classify MT errors. Most MT error classification systems in the literature are designed for use by potential MT users, and thus focus on system comparison and on ways to measure translation quality from an end-user viewpoint (Flanagan, 1994; White et al., 1994).

The purpose of the MT error classification system for the TCTool is radically different. Instead of having translation and linguistic experts make judgements for the consumption of end users, we have non-expert users making the judgements that should allow us to obtain more information about what might be the cause of a translation error.

Automated MT evaluation methods developed recently (Papineni et al., 1998), on the other hand, are based on ngram precision compared to a reference translation, and even though they are most useful for managers and developers to know whether a change on the MTS has any impact in accuracy (as well as to do system comparisons), these measures do not give any insight about what the developer might need to do to improve the system.

What we're trying to elicit from the user, is precisely what's missing from existing evaluation methods.

For this reason, we need to think of MT error classification in a completely different way, and we need to find the balance between simplicity and informativeness. Naive users have to be able to understand the different error types and classify errors accurately, and at the same time, we have to obtain the most information that they can possibly give us, in order to be able to automatically refine translation rules.

Therefore, a big aspect of this research is to find the right balance between these two parameters and figure out what is the best MT error classification possible that will allow users to be maximally accurate and informative.

The MT error classification used in the first English-Spanish users study has 9 categories: wrong word order, wrong sense of the word, wrong agreement (number, person, gender, tense), wrong form of the word, incorrect word and no translation. It is meant to see how well users can tell

the different error types apart. For a brief explanation of the error types with examples, see (Font Llitjós, 2004).

Clearly, this classification could be much finer grained. For example, currently "wrong form" includes things like case and part-of-speech, and if we find that users are able to tell these two kinds of errors apart, we will add them as separate error types.

However, the first user study presented below indicates that a coarser grained classification might actually be better for this task.

Either way, we expect this classification to change as we observe what users do and analyze their behavior when using the TCTool.

4. The Translation Correction Tool: TCTool

The TCTool is thought of as a user-friendly way to get bilingual users to easily evaluate and correct machine translated sentences online from an Internet browser.

The TCTool presents the user with a sentence in the source language with up to five translations in the target language, and asks them to either check all the correct translations, or, if none of the alternative translations is correct, to to fix the best translation with the least amount of corrections possible.

In this context, the best incorrect translation is the one requiring the least number of changes to render the same meaning as the original sentence, in a grammatically correct and fluent target language sentence.

The most important aspect of the TCTool and this user study is that user feedback is not only used to improve the translation at hand, but it is critical for the refinement of the translation rules and will be used to improve the MTS at its core. For this reason, it is very important that users only correct what is strictly necessary to obtain a correct translation of the original sentence from the given translation.

Translations have two components: the words in the target language and the alignments from the sl to the tl. The alignments indicate the word-to-word correspondence, namely what word in the sl translates as a word in the translation sentence.

The TCTool instructions explain what it means to correct a translation minimally, and a 23-page long TCTool tutorial illustrates how users can do this using the TCTool. The tutorial shows the possible actions to correct a translation with 4 example sentences. Figures 1 and 2 below show the TCTool interface before and in the middle of the correction of the first sentence. See (Font Llitjós, 2004) for more details.

5. Interface design and implementation

The TCTool interface is designed to abstract away as much as possible from what is happening inside the MTS, and to allow users to correct errors at a relatively high level.

Since we are aware of the intrinsic difficulty of the task at hand, we tried to choose an interface that is fun to play with, and that will guide users on how to correct a translation and, at the same time, will give them some flexibility.

This was the main reason we used JavaScript 1.2. for the application that allows users to correct a sentence, while all the other cgi scripting is in Perl.

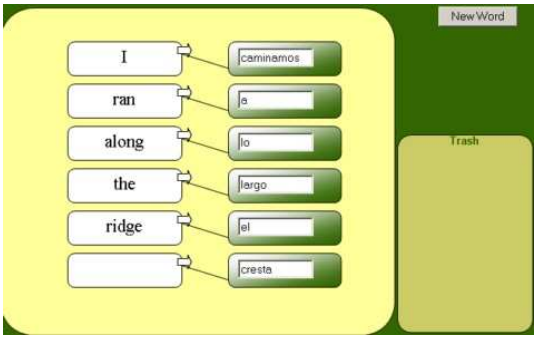

Figure 1: Example of initial screen with incorrect translation.

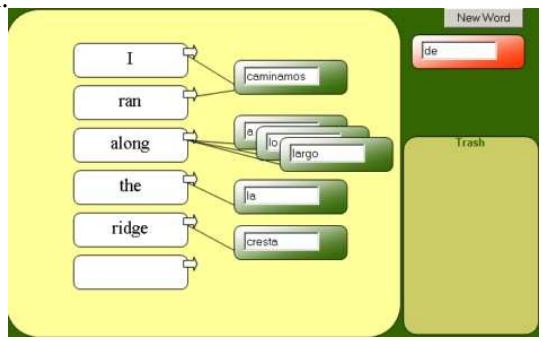

Figure 2: Example of screen with sentence in the process of being corrected.

As shown in figures 1 and 2 above, when correcting a sentence, the user is presented with two columns of blocks, each block containing a word. The sl sentence is displayed on the left column, and the tl sentence is displayed on the right. The alignments between the source and target sides are originally extracted from the translation rule(s) that generated the sentence and appear as arrows from sl sentence to tl sentence.

- edit a word
- add a word
- delete a word
- drag a word into a different position
- add an alignment
- delete an alignment

Figure 3: Possible actions to correct a sentence.

There are six basic operations users can do to correct a sentence, as shown in figure 4. The first one has a set of error types associated with it (the ones described in section 3 above), and the user is asked to pick all the one(s) considered to be the cause of the error that they are correcting.

5.1. Implementation details

The TCTool takes as input a file with all 32 sentences, their translations and alignments, and presents them to the user one sl sentence at a time.

The current implementation of the TCTool makes the assumption that if a translation is correct, the alignments for that translation are also correct.

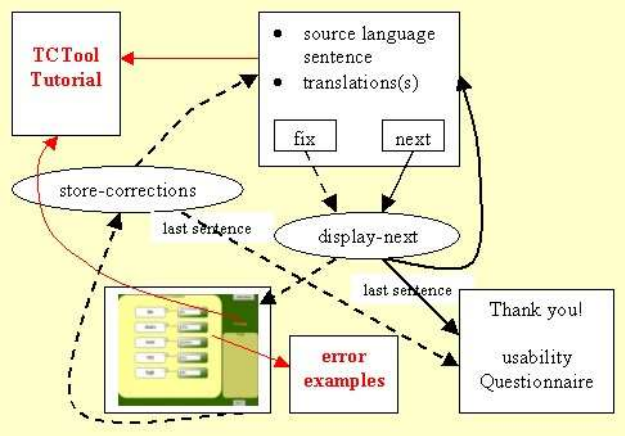

Figure 4: TCTool simplified data flow diagram. Bubbles represent Perl cgi scripts; the colorful square represents the JavaScript application; other squares, HTML pages generated by the scripts; discontinuous arrows represent the flow if the translation is not correct and user decides to fix it; continuous black arrows show the flow if translation is acceptable; everything in red are help pages.

The version for the English-Spanish user study has five cgi scripts in Perl, one in JavaScript, and they produce a total of eight different HTML pages. In the simplified data flow diagram below (figure 6), you can see how the core of the TCTool works.

6. TCTool English-Spanish user study

The purpose for the first TCTool user study is twofold. First, to assess how well the current TCTool interface and the MT error classification work for the purpose of detecting and correcting machine translation errors. Second, to evaluate how well people can detect and correct machine translation errors in general, and whether users can use the TCTool without any training or supervision.

This first evaluation presents users with 32 simple English sentences chosen from the corpus designed to elicit various linguistic phenomena (Probst et al., 2001) to display a wide variety of MT errors.

There are many different ways to translate a sentence, and we ask users to pick the one that will allow them to make the least number of changes possible to render the original meaning in a grammatically correct and fluent way.

The concept of minimal correction is really important here, since we are aiming at using such corrections to refine the underlying translation rules directly. If users changed a translation completely, so that it is the way they'd prefer to translate the source sentence, that would not be of much use to our ultimate goal.

An example of sentence used in the user study is the source language sentence "the chairs were very high". After running it through the transfer engine and post-processing, we obtain the following:

```
sl: the chairs were very high
tl: las sillas estaban muy alto
al: ((1,1),(2,2),(3,3),(4,4),(5,5))
```

In this case, users need to correct the translation so that the predicative adjective "alto" agrees with the subject of the sentence in Spanish, "las sillas", which is feminine plural. To do that, users need to edit the last word "alto", change it to "altas" and then mark two causes of errors: wrong number and gender agreement.

It would also be correct to translate the SL sentence as "las sillas eran muy altas" or "las sillas estaban muy arriba", but that would involve making more changes to the translation than strictly necessary, since "las sillas estaban muy altas" is already a correct translation of the sl sentence.

6.1. MTS configuration

The MTS used for this user study consists of a manually written English to Spanish grammar with 12 translation rules (2 S rules, 7 NP rules and 3 VP rules) and 442 lexical entries, which were designed to translate the first 400 sentences of the elicitation corpus mentioned above.

6.2. Target users

The target users for the first user study were native speakers of Spanish with good knowledge of English. They are not assumed to know anything about linguistics nor translation.

Since the our MTS is ultimately going to have a low-density language as tl, we can't expect users to be computer literate either, and that is the reason much effort was put into designing the TCTool interface.

Before staring the evaluation, users are asked to answer a few classification questions including where they were born and raised and the level of education.

6.3. Actual user statistics

There were 29 users who completed the evaluation. Most users were from Spain (83%).

Two thirds of the users did not have any background in Linguistics, 75% had a graduate degree and 25% of the users had a Bachelor's degree.

On average, users took an hour and a half to evaluate the 32 translation sets and fix 26.6 translations, about 3 minutes per sentence. But there was a significant fluctuation among users, the duration range being [28min-4:18hours].

6.4. Gold standard: measuring user accuracy

In order to be able to measure user accuracy in detecting and classifying errors, we need to establish exactly what is the minimum number of errors and corrections needed per translation. For that, we created a gold standard which determines what are the least number of errors that must be corrected and what are the error types, if applicable.

To measure accuracy, i.e. how close are users from the gold standard, we looked at precision, recall and F1 measure.

In this context, precision is a measure of the proportion of errors that the user fixed correctly (# errors detected correctly / # errors detected). And since we are mostly interested in the accuracy of users when telling us what is the type of the error, we also estimated the precision in which users checked the right error type.

Recall is a measure of the proportion of the errors in the translations that the user detected (# errors detected correctly / # errors there are in gold standard).

Usually there is a trade-off between precision and recall, and the F1 measure is an even combination of the two. It is defined as [2*p*r/(p+r)]. All three measures fall in the range from 0 to 1, with 1 being the best score.

For the TCTool, we are interested in high precision at the expense of lower recall. In other words, we'd like no false positives (users correcting something that is not strictly necessary), and we don't worry so much about having false negatives (errors that were not corrected).

6.4.1. For a subset of 10 users

We measured precision, recall and F1 for 10 (of the 29) users with the following characteristics: all of them were from Spain, only two of them had Linguistics background, and 2 a Bachelor's degree, 5 a Masters and 3 a PhD.

	precision	recall	F1
error detection	0.896	0.894	0.895
error classification	0.724	0.715	0.719

Table 1: Average accuracy mesaures for 10 users and 32 sentences

Users did not always correct translations in the same way. Most of the time, when the final translation was not like the gold standard, it was still correct, and some times it was even better. On average, users only produced 2.5 translations that were worse than the gold standard (out of the 26.6 that they corrected). Surprisingly, users got most alignments correctly.

We run three test-drives with different profile users (from very distinct geographical areas), which indicated that users got familiar with the tool after the 2nd to 9th sentence.

6.5. Usability questionnaire

At the end of the evaluation, users filled out a usability questionnaire, and they indicated that they thought the TCTool was user-friendly (82%), but that the alignment representation could be improved (67%).

All users said that it is easy to determine if a sentence translation is correct, but the number of users who felt that determining the source of errors is easy goes down by 12%. In general, users felt that actually fixing the translations was a bit harder.

7. Conclusions

Rule-based transfer MTS are inherently limited. If the rules are written manually, no matter how many rules there are, coverage can always be increased. If they are automatically learned, they might contain either overgeneralizations or lack of constraints. Either way, in face of unseen examples, the translation rules will need to be refined to account for the new data.

The TCTool is an online tool that allows us to get guided and structured user feedback on translations generated by our transfer MTS, with the ultimate goal of automatically improving the translation rules.

This first user study shows that users can detect errors with high accuracy (89% of the time), but have a harder time classifying error given the MT error classification above. Users assigned the right error type only 72% of the time.

In general, most of the problems users had were due to not having read the instructions and the tutorial.

8. Future Work

The next version of the TCTool will have a dynamic tutorial, which will provide appropriate information when users try to do something on the same screen they are correcting the sentence.

Error detection accuracy is rather high, but for rule refinement purposes, what we are really interested in is high error classification accuracy. In the future, we will work on developing an MT error classification that results into higher error classification accuracy, specially into higher precision.

The main reason of being for the TCTool is to be able to extract user feedback and use it to improve the underlying translation grammar. The next step is to analyze user feedback to see how we can automatize the rule refinement process.

9. Acknowledgements

We would like to thank Kenneth Sim and Patrick Milholl for the implementation of the JavaScript.

10. References

Font Llitjós, A., 2004. The translation correction tool: English - Spanish user study. url: http://avenue.lti.cs.cmu.edu/aria/spanish/ http://avenue.lti.cs.cmu.edu/aria/spanish/tutorial.html http://avenue.lti.cs.cmu.edu/aria/spanish/error-examples.html

Probst, K., Levin, L., Peterson, E., Lavie, A. and Carbonell, J., 2003. MT for Resource-Poor Languages Using Elicitation-Based Learning of Syntactic Transfer Rules. To appear in (has it appeared yet?): Machine Translation, Special Issue on Embedded MT. 2003.

Probst, K., Brown, R., Carbonell, J., Lavie, A. Levin, and L., Peterson, E., 2001. Design and Implementation of Controlled Elicitation for Machine Translation of Low-density Languages. *Proceedings of the MT2010 workshop at MT Summit 2001*.

Papineni, K., Roukos, S., and Ward, T. Maximum Likelihood and Discriminative Training of Direct Translation Models. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP-98)*, 189-192.

Flanagan, M., 1994. Error Classification for MT Evaluation. *Proceedings of AMTA 94*, 65-72.

White, J.S., O'Connell, T. and O'Mara, F., 1994. The ARPA MT Evaluation Methodologies: Evaluation, Lessons, and Future Approaches. *Proceedings of AMTA 94*, 193-205.