

A Web-Based Interactive Computer Aided Translation Tool

Philipp Koehn

School of Informatics
University of Edinburgh
pkoehn@inf.ed.ac.uk

Abstract

We developed **caitra**, a novel tool that aids human translators by (a) making suggestions for sentence completion in an interactive machine translation setting, (b) providing alternative word and phrase translations, and (c) allowing them to post-edit machine translation output. The tool uses the Moses decoder, is implemented in Ruby on Rails and C++ and delivered over the web.

1 Introduction

Today's machine translation systems are mostly used for inbound translation (also called assimilation), where the reader accepts lower quality translation for instant access to foreign language text. The standards are much higher for outbound translation (also called dissemination), where the reader is typically an unsuspecting customer or citizen who is seeking information about products or services, and human translators are required for high-quality publication-ready translation.

While machine translation has made tremendous progress over the last years, this progress has made little inroads into tools for human translators. Although it has become common practice in the industry to provide human translators with machine translation output that they have to post-edit, typically no deeper integration of machine translation and human translation is found in translation agencies.

An interesting approach was pioneered by the TransType project (Langlais et al., 2000). The machine translation system makes sentence completion predictions in an interactive machine translation setting. The users may accept them or override them by typing in their own translations, which triggers new suggestions by the tool (Barachina et al., 2009).

But also other information that is generated during the machine translation process may be useful for the human translator, such as alternative translations for the input words and phrases.

We are at the beginning of a research program to explore the benefits of these different types of aid to human translators, analyze user interaction behavior, and develop novel types of assistance. To have a testbed for this research, we developed an online, web-based tool for translators.

2 Overview

Caitra is implemented in Ruby on Rails (Thomas and Hansson, 2008) as a web-based client-server architecture, using Ajax-style Web 2.0 technologies (Raymond, 2007) connected to a MySQL database-driven back-end. The machine translation back-end is powered by the open source Moses decoder (Koehn et al., 2007). The interactive machine translation prediction code is implemented in C++ for speed. The tool is delivered over the web to allow for easier user studies with remote users, but also to expose the tool to a wider community to gather additional feedback. You can find caitra online at <http://www.caitra.org/>

Caitra allows the uploading of documents using a simple text box. This text is then processed by a back-end job to pre-compute all the necessary data (machine translation output, translation options, search graphs). This process takes a few minutes.

Finally, the user is presented with an interface that includes all the different types of assistance. Each may be turned off, if the user finds it distracting. The user translates one sentence at a time, while the context (both input and user translation, including the proceeding and following paragraph) is displayed for reference.

In the next three sections, we will describe each type of assistance in detail.

3 Interactive Machine Translation

The idea of interactive machine translation has been greatly advanced by work carried out in the TransType project (Langlais et al., 2000), with the focus on a sentence-completion paradigm. While the human translator is still in charge of creating

(1) Paul Newman le magnifique >>



Figure 1: **Interactive Machine Translation.** Caitra uses the search graph of the machine translation decoder to suggest words and phrases to continue the translation.

the translation word by word, she is aided by a machine translation system that interactively makes suggestions for completing the sentence, and updates these suggestions based on user input. The scenario is very similar to the auto-completion function for words, search terms, email addresses, etc. in modern office applications.

See Figure 1 for a screenshot of the incarnation of this method in our translation tool. The user is given an input sentence and a standard web text box to type in her translation. In addition, caitra makes suggestions about the next word (or phrase) to be added to the translation. The user may accept this (by pressing the TAB key), or type in her own translation. The tool updates the prediction based on the user input.

The predictions are based on a statistical machine translation system. Given the input and the partial translation of the user (called the prefix), the machine translation system computes the optimal translation of the input sentence, constrained by matching the user input. This translation is provided to the user in form of short phrases (mirroring the underlying phrase-based statistical translation model).

In contrast to traditional work on interactive machine translation, the displayed suggestions consist of only very few words to not overload the reading capacity of the user. We have not yet carried out studies to explore the optimal length of suggestions, or even when not to provide suggestions at all, in cases when they will be most likely useless and distracting.

We store the search graph produced by the machine translation decoder in a database. During the user interaction, we quickly match user input against the graph using a string edit distance measure. The prediction is the optimal completion path that matches the user input with (a) minimal

Paul	Newman	le	magnifique
Paul	Newman	the	wonderful
Mr	Newman ,	the	magnificent
Mr Paul	Newman here	the	wonderful
as Paul	Committee		beautiful
another	Newman , who speaks		magnificent
with Paul			the splendid
, Paul			the excellent
of Paul			the beautiful
work of Paul			it
the words of Paul			great

Figure 2: **Translation Options.** The most likely word and phrase translation are displayed alongside the input words, ranked and color-coded by their probability.

string edit distance and (b) highest sentence translation probability. This computation takes place at the server and is implemented in C++.

While caitra only displays one phrase prediction at a time, the entire completion path is transmitted to the client. Acceptance of a system suggestion will instantly lead to another suggestion, while typed-in user translations require the computation of a new sentence completion path. This typically takes less than a second.

Preliminary studies suggest that users accept up to 50-80% of system predictions, but obviously this number depends highly on language pair and difficulty of the text.

4 Options from the Translation Table

Phrase-based statistical machine translation methods acquire their translation knowledge in form of large phrase translation tables automatically from large amounts of translated texts (Koehn et al., 2003). For each input word or input word sequence, this translation table is consulted for the most likely translation options. A heuristic beam search algorithm explores these options and their ordering to find the most likely sentence translation (which takes into account various scoring functions, such as the use of an n-gram language model).

These translation options may also be of interest to the user, so we display them in our translation tool caitra. See Figure 2 for an example. For instance, the tool suggests for the translation of the French *magnifique* the English options *wonderful*, *beautiful*, *magnificent*, and *great*, among others. The user may click on any of these phrases and

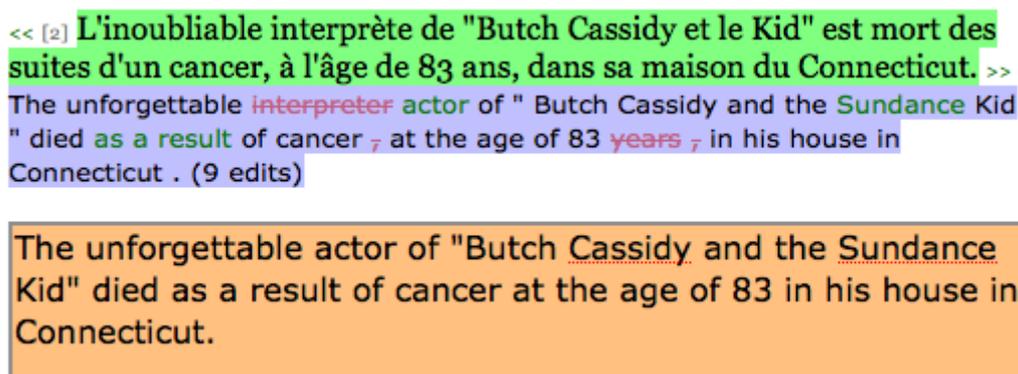


Figure 3: **Post-Editing Machine Translation.** Starting with the sentence translation of the machine translation system, the user post-edits and the tool indicates changes.

they are added into the text box. The user may also just glance at these suggestions and then type in the translations herself.

The options are color-coded and ranked based on their score. Note that since these options are extracted from a translated corpus using various automatic methods, often inappropriate translations are included, such as the translation of *Newman* into *Committee*.

For each translation option a score is computed to assess its utility. This score is the (i) future cost estimates of the phrases (ii) plus the outside cost estimates for the remaining sentence (iii) minus the future cost estimate for the full sentence. This number allows the ranking of words vs. phrases of different length. The ranking of the phrases never places a lower scoring option above a higher scoring option. The absolute score is used to color code the options. Up to ten table rows are filled with options.

Since the user may click on the options, or may simply type in translations inspired by the options, it is not straight-forward to evaluate their usefulness. We plan to assess this by measuring translation speed and quality. Experience so far has shown that the options help novice users with unknown words and advanced users with suggestions that are not part of their active vocabulary. It may be possible that these options even allow users that do not know the source language to create a translation, as in work done by Albrecht et al. (2009).

5 Post-Editing Machine Translation

The addition of full sentence translation of the machine translation system is trivial compared to the other types of assistance. When a user starts a new

sentence using this aid, the text box already contains the machine translation output and the user only makes changes to correct errors.

See Figure 3 for an example. Caitra also compares the user's translation in form of string edit distance against the original machine translation. This is illustrated above the text box, to possibly alert the user to mistakenly dropped or added content.

6 Key Stroke Logging

Caitra tracks every key stroke and mouse click of the user, which then allows for a detailed analysis of the user's interaction with the tool. See Figure 4 for a graphical representation of the user activity during the translation of a sentence. The graph plots sentence length (in characters) against the progression of time. Bars indicate the sentence length at each point in time when a user action takes place (acceptance of predictions are red, DEL key strokes purple, key strokes for cursor movement grey, and key strokes that add characters are black.)

In the example sentence, the user first slowly accepted the interactive machine translation predictions (second 0-12), then more rapidly (second 12-20), followed by a period of deletions and typing that did not make the translation longer (second 20-30). After a short pause, predictions were accepted again (second 33-40), followed by deletions and typing (second 40-57).

We are currently carrying out user studies to not only compare the productivity improvements gained by the different types of help offered to the user, but also to identify, categorize and analyze the types of activities (such as long pauses,

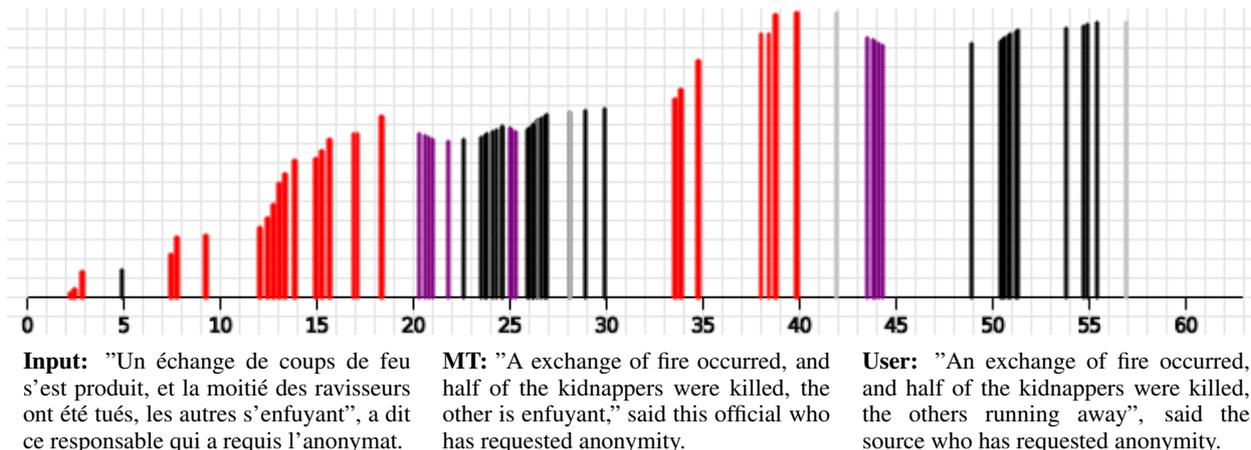


Figure 4: **User Activity.** The graph plots the time spent on translation (in seconds, x-axis) against the length of the sentence (y-axis) with color-coded activities (bars). For instance, at the interval second 2–3, three interactive machine translations predictions were accepted.

slow typing, fast typing, clicks on options, acceptance of predictions) to gain insight into the type of problems in (computer aided) human translation and the time spent to solve these problems.

7 Conclusions

We described the new computer aided translation tool *caitra* that allows us to compare industry-standard post-editing, the interactive sentence completion paradigm, and other help for translators. The tool is available online at the URL <http://www.caitra.org/>.

We will report on user studies in future papers.

8 Acknowledgments

This work was supported by the EuroMatrix-Plus project funded by the Europea Commission (7th Framework Programme). Thanks to Josh Schroeder for help with Ruby on Rails.

References

- Albrecht, J., Hwa, R., and Marai, G. E. (2009). Correcting automatic translations through collaborations between mt and monolingual target-language users. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*.
- Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A., Ney, H., Tomás, J., Vidal, E., and Vilar, J.-M. (2009). Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C. J., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

- Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase based translation. In *Proceedings of the Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

- Langlais, P., Foster, G., and Lapalme, G. (2000). Transtype: a computer-aided translation typing system. In *Proceedings of the ANLP-NAACL 2000 Workshop on Embedded Machine Translation Systems*.

- Raymond, S. (2007). *Ajax on Rails*. O'Reilly.

- Thomas, D. and Hansson, D. H. (2008). *Agile Web Development with Rails: Second Edition, 2nd Edition*. The Pragmatic Programmers, LLC.