Meta
Journal des traducteurs
Translators' Journal

META

A New Computational Tool for Analyzing Translation Processes: The TransCorrect Project

Jean-Pierre Colson

Volume 50, numéro 2, avril 2005

Processus et cheminements en traduction et interprétation
Processes and Pathways in Translation and Interpretation

URI : https://id.erudit.org/iderudit/011001ar
DOI : https://doi.org/10.7202/011001ar

Aller au sommaire du numéro

Éditeur(s)

Les Presses de l'Université de Montréal

ISSN

0026-0452 (imprimé)
1492-1421 (numérique)

Découvrir la revue

Citer cet article

Colson, J.-P. (2005). A New Computational Tool for Analyzing Translation Processes: The TransCorrect Project. *Meta*, *50*(2), 573–579. https://doi.org/10.7202/011001ar

Résumé de l'article

Le projet TransCorrect représente une première tentative de correction automatisée de la traduction humaine dans le contexte de l'enseignement supérieur. Il se fonde sur une série de nouvelles techniques héritées de la linguistique computationnelle de corpus, mais également sur l'expérience pédagogique de l'auteur. Le programme a été spécialement conçu pour l'enseignement de la traduction à l'université. Tout en offrant aux étudiants un retour immédiat à chaque tentative de traduction, le projet TransCorrect permet également à l'enseignant d'aborder sous un angle différent la complexité des processus de traduction.

A New Computational Tool for Analyzing Translation Processes: The TransCorrect Project

JEAN-PIERRE COLSON
Université catholique de Louvain, Louvain-la-Neuve, Belgique
colson@fltr.ucl.ac.be

RÉSUMÉ

Le projet TransCorrect représente une première tentative de correction automatisée de la traduction humaine dans le contexte de l'enseignement supérieur. Il se fonde sur une série de nouvelles techniques héritées de la linguistique computationnelle de corpus, mais également sur l'expérience pédagogique de l'auteur. Le programme a été spécialement conçu pour l'enseignement de la traduction à l'université. Tout en offrant aux étudiants un retour immédiat à chaque tentative de traduction, le projet TransCorrect permet également à l'enseignant d'aborder sous un angle différent la complexité des processus de traduction.

ABSTRACT

The TransCorrect project is a first attempt to automate the correction of human translation within the context of higher education. It is based on a number of new techniques derived from computational corpus linguistics, but also on the author's pedagogical experience. The program was specially created for the teaching of translation at university level. While offering the students an immediate feedback to their translation attempts, the TransCorrect project also enables the teacher to have an additional insight into the complex translation processes.

MOTS-CLÉS/KEYWORDS

automatic translation evaluation, translation processes, computational linguistics, language automation, translation and technology

Introduction

This article reports the first results of a research project designed at improving the translation teacher's control over the performance of his students by means of a computational approach. It goes without saying that the program presented here has no other pretension than to serve as an additional tool, which in no way replaces or undermines the importance of traditional methods.

The starting point of the project was simply my desire to somehow vary my teaching of translation by using the computer room and by offering the students an immediate feedback on the screen for their translation attempts, while leaving time for a more personal tutoring. At first sight, the whole idea seemed somehow unrealistic. However, a first experiment involving the translation of a single text from Dutch into French was carried out in March 2004 in Brussels, and the students were so enthusiastic about it that I decided to devote hundreds of hours to the development of a general executable that might be used by any translation teacher in any language combination. This resulted in the creation of two programs within the TransCorrect project: the student version, an open program that can be used with

any text, and the master program, with which translation teachers can create their own evaluation and comments.

1. Theoretical background

The growing success of corpus linguistics (initiated by Sinclair, 1991) has led researchers to develop ever more sophisticated techniques for studying linguistic data and language performance by learners. A well-known example is the concordancer, an instrument used by many linguists and language students around the world.

While corpus linguists pleaded for a return to the accurate analysis of objective data (as opposed to the generative approach of the Chomskian school), computational linguistics also created a number of new algorithms and innovative techniques based on new programming languages. It is therefore no wonder that a new discipline is now emerging from those two directions: computational corpus linguistics, whose field of investigation covers various aspects of language use, including translation.

One of the most currently used programming languages among corpus linguists is Perl, the Practical Extraction and Report Language, created by Larry Wall in the mid-1980s. One of the great advantages of Perl is its ability to manipulate text by means of powerful algorithms, among which a new version of Henry Spencer's V8-subroutines, which came to be known as *regular expressions*. Perl's regular expressions are so powerful and efficient that most programming languages (Delphi, C++, C#, Java, Visual Basic, etc.) have taken them over. Perl is an interpreted language, which means that its instructions are very short, all the tedious work being done by its interpreter running in the powerful C language. For a linguist, this implies that you don't have to know a lot about programming before writing a Perl script that will on the other hand have all the power of a real executable written in C.

Regular expressions were not created in the first place for linguists, but for manipulating and checking all kinds of chains, such as codes, passwords, bank card numbers, statistical results, Web pages and so on. However, their incredible power (based on the mathematical theory of non-finite automatons) quickly made them a favourite instrument of corpus linguists. Their very compact syntax (for a full overview, see Friedl 2002) makes it possible to take all kinds of linguistic variants into account. For instance, the regular expression (here in Perl syntax):

```
/fills?(ed)?/i
```

recognizes *fill*, *fills*, *filled* in small or in capital letters, or with a combination of both.

Regexes (regular expressions) are based on just one among many algorithms used by computational linguists in order to analyze texts. Search algorithms, for instance, have been improved in such a way that a huge corpus can be searched in a handful of seconds. All these tools put together not only opened new research lines to corpus linguistics, but also to automatic or computer aided translation.

The BLEU project (Papineni et al. 2001), for instance, used a number of algorithms in order to evaluate the results of automatic translation programs. The Bleu score is obtained by comparing the source text and the target text produced by the program, on the basis of statistical criteria. According to the authors, there is a high correlation between the bleu score and the evaluation of the automatic translations carried out by a control group of native speakers.

Naturally, linguists tend to be sceptical when computational scientists try to convince them that language can be reduced to statistical formulas and purely quantitative data. Indeed, the semantic aspect of language is known to pose many problems to corpus and computational theories (see Čermák 2002). For instance, a frequent co-occurrence of 2 words (a *bigram* in statistical terms) can correspond to quite different grammatical or idiomatic situations. A *sleeping policeman* is a rather frequent *bigram* on the Internet, but only a man using his semantic ability can distinguish between a sleeping man and a traffic hump (French: *un ralentisseur, un casse-vitesse*).

In other words, a fully automated analysis of a source and target text, or of the production of students when translating texts, is not very realistic – at least today. This is the reason why the TransCorrect project is based on a combination of automated features and a manual evaluation.

2. Brief presentation of the TransCorrect project

2.1. The teacher's master program (MasterCorrect)

The master program enables any translation teacher working in any language combination (at least for European languages, so far) to create an automated correction of translations produced by the students. As illustrated by the screenshot (picture 1), the translation teacher can divide the source text into a number of sections (10 in the present version), corresponding roughly to three or four ordinary sentences. Picture 1 illustrates the translation of a fragment of text from Dutch into French.

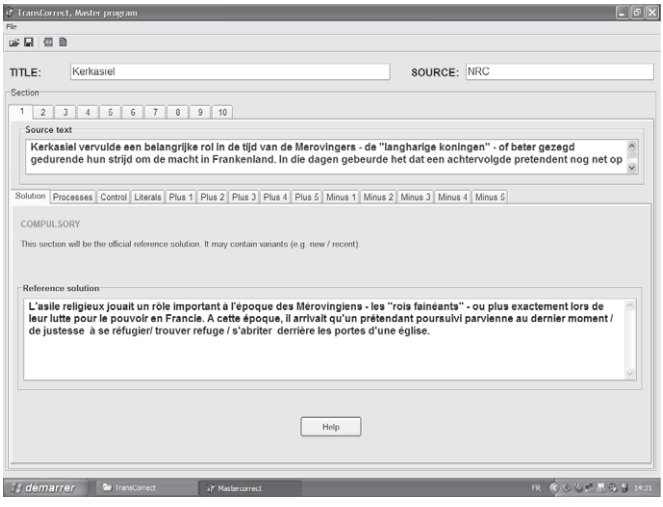
The teacher just has to fill in the blanks: title, source, source text, and then a number of windows that can be activated by a click (a Page Control in Windows terminology). In picture 1, the first page is active, and it is the one devoted to the reference solution. Here, the teacher must enter the best solution(s) for each section of the source text. The other parameters are:

- Processes: the teacher enters a brief description of all translation processes and techniques involved in the translation of the source sentence(s);
- Control: one or more control elements that must be present in the student's translation (if the student types *blablabla* or nothing at all, the program will recognize an invalid solution);
- Literals: a number of literal elements (e.g. New York, John, today, three) that will necessarily remain unchanged in the target sentence(s), or receive only one possible translation;
- Five positive evaluation criteria (labelled *Plus 1*, *Plus 2* etc.): these refer to a maximum of five chains (a word, a construction with possible variants) that might be found in the student's translation. Each time, the teacher can enter a comment that will appear on the student's program. This is illustrated by the example in Picture 2: if the student's translation of the sentence includes *remplissait* or *remplit* or *a rempli*, the comment at the bottom of the window will be displayed on the student's computer. It is worth mentioning that teachers having a basic mastery of *Perl 5 regular expressions* can replace the three possible variants by just one regex: '(a\s)?remplit?(ssait)?
- Five negative evaluation criteria (labelled *Minus 1*, *Minus 2* etc.): these work precisely in the same way as the five positive ones, but they detect the absence of a maximum of five chains (a word, a construction with possible variants) per source sentence.

As complex algorithms are involved in the treatment of the student's answer, the number of variants or of positive and negative elements cannot be infinitely

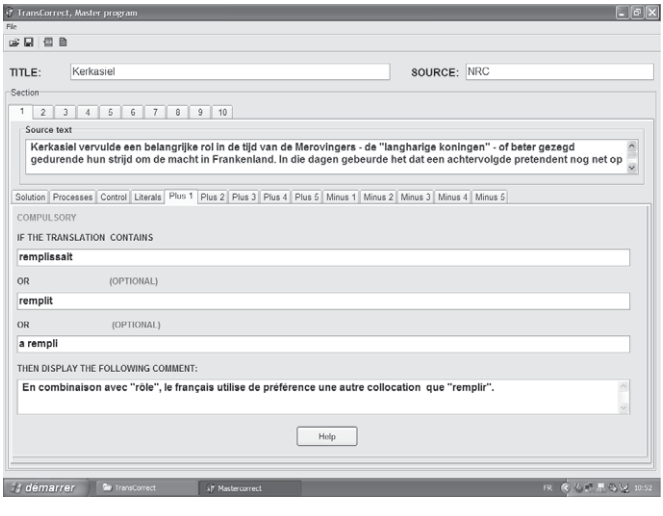
PICTURE 1

Loading a text in the master program



PICTURE 2

Entering a teacher's comment



extended, but the present version of TransCorrect already offers a sufficient coverage of the student's output.

2.2. The student's program (TransCorrect)

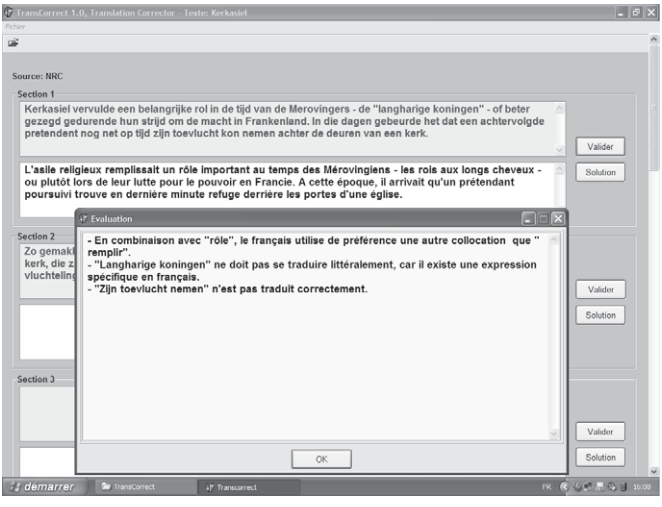
Picture 3 gives an overview of the user end of the program: another executable that works in combination with the first one, and is primarily intended for translation students working in a computer room.

The student's program is empty when you start it (just like an MS Word program before you open a document), but it can load any text prepared by the master program. These texts have the form of special files containing not only all the sentences of the text to be translated, but also all the possible mistakes, the solutions, and the teacher's comment. Needless to say, all this information is not directly accessible by opening the files, as it is encrypted according to complex algorithms. Only the student's program is able to achieve the decoding in a flash, as it is pre-programmed with all the algorithms and passwords. It is also possible to encrypt texts for a specific group of students, so that another group will not be able to work with it.

Picture 3 shows, at the user end, the event corresponding to picture 2 above. The student has loaded the text, and has entered his translation, which triggers a comment from the program (here in the French version).

PICTURE 3

Typing a translation and receiving an automated evaluation



As stated above, the student only gets an evaluation of his translation if a number of elements are present in his target text. These elements have been pre-programmed by the teacher using the master program. Clicking on *Solution* is also subject to the same restrictions. It is therefore impossible to get the solution if one does not try a few translations first.

When the students do ask for the solution, they not only get the text typed by the teacher, but also a list of translation techniques that should have been used for this specific sentence. All those comments are automatically generated by the program, but they just reflect the teacher's point of view.

Another important technical feature of TransCorrect is that every translation attempt by the students is recorded. Each time they click on *Enter* (or *Valider* in the French version presented in picture 3), the program automatically saves the typed sentences to a text file. Afterwards, the teacher can therefore compare all the reactions of a given student with the automated comment.

3. An additional insight into translation processes

The TransCorrect project did not originate from purely statistical or computational research, but was rather meant to meet a pedagogical and methodological demand. Teaching translation and interpreting at university level is an enriching but very complex task. The multimedia revolution has provided translation teachers and students with a lot of new tools. However, there was so far no computer program specifically designed for the evaluation of the translations produced by students and for analyzing the processes and techniques involved.

Once again, TransCorrect does not claim to offer anything like a perfect computational solution to all the problems involved, but it is a first attempt at alleviating the translation teacher's task, while providing him with information that he would impossibly have derived from his traditional teaching.

From the teacher's point of view, the TransCorrect project requires first of all a few hours work on MasterCorrect (from 2 to 5 for an average text). However, it is important to stress that teachers may decide to create many texts with minimal information, or to spend more time on a fine-tuned analysis. The teacher can make his automated evaluation program as general or as accurate as he pleases. He may work for a few hours on a text, but the result is a compiled text that he will be able to use for many years. So far, the program has only been tested on a group of twenty students working in the computer room of the Marie Haps Institute in Brussels. This first experiment clearly enabled the teacher to spend more time observing the translation strategies used by students, answering specific questions, and solving individual problems. Contrary to the traditional classroom or seminar situation of translation courses, this new methodology allows clearly for a more personal tutoring.

As stated before, the students having taken part in the first experiment were enthusiastic. For the first time, they were able to work for several hours (by periods of 2 hours) on a text, while receiving an instant evaluation of all their translation attempts. It must be clear, by now, that this automated evaluation is nothing else than a summary of the teacher's analysis. The formulation of the solutions and comments is, in other words, quite similar to the teacher's traditional teaching and there is no impression of being confronted with some kind of robot telling you how to translate. Students are active all the time, and have therefore the feeling of making faster progress.

Maybe the most important feature of TransCorrect is its ability to trace down translation processes in a new way. Again, this program should be used in addition with traditional methods, and this particularly holds for the investigation into translation processes.

As all the translation attempts entered by the students are automatically recorded by the program, the teacher is able to add this information to his pedagogical approach. For each student, he can check the number of translation attempts, their quality, but also their adaptation to the comment that was yielded by the computer. A discussion with the students is also possible on the basis of this information. To take a simple example, a typical comment generated by the program is a remark about some kind of stylistic problem, based on the recognition by the algorithm of one or more stylistically weak constructions. When students are told that there is a stylistic problem in their translation, their reactions may vary enormously. Some

students, for instance, will go on trying four or five new translations, while others will directly identify the source of the problem. This kind of stylistic uncertainty is rather difficult to pinpoint with a traditional method, because the teacher usually has no dialogue with the student while he is creating his translation. Obviously, not everything is solved by taking recourse to TransCorrect, but the first experiment suggests that a better understanding of the translation processes is made available.

4. A few final remarks

The recent developments of computational corpus linguistics have made new techniques available for the manipulation of written language. However, no attempt had been made so far to design practical tools for the correction of human translation, especially within the context of higher education. The TransCorrect project should be seen as a first step in this direction, with a number of limitations but also some original features.

Contrary to the mainstream of computational linguistics, TransCorrect does not use *brutal force*, i.e. a fully automated analysis of the target text, but is based on the translation teacher's competence and on his ability to summarize a number of translation problems. This does not mean that a combination of this method and of full automation should be excluded. TransCorrect 1.0. is just an experiment, and it is open to adaptations and improvements. All suggestions, comments or questions are welcome (jp.colson@ilmh.be).

REFERENCES

Colson, J.-P. (1992): "Phraséologie et terminologie en traduction et en interprétation," *Le Langage et l'Homme* 27, p. 119-120.

Colson, J.-P. (1995): "Quelques remarques sur l'enseignement de la phraséologie aux futurs traducteurs et interprètes," *Le Langage et l'Homme* 30, p. 147-156.

Colson, J.-P. (2003): "Corpus Linguistics and Phraseological Statistics: A Few Hypotheses and Examples," *Flut von Texten – Vielfalt der Kulturen. Ascona 2001 zur Methodologie und Kulturspezifik der Phraseologie* (Burger, H., Häcki Buhofer, A. & G. Gréciano, eds), Baltmannsweiler, Schneider Verlag, p. 47-59.

Colson, J.-P. (in press): "The World Wide Web as a Corpus for Set Phrases," *Phraseologie / Phraseology, Handbooks of Linguistics and Communication Science* (Burger, H., Dobrovol'skij, D., Kühn, P. & N. Norrick, eds.), Berlin, New York, Mouton de Gruyter.

Colson, J.-P., Granger, S. and L. Beheydt (1999, eds.): *Linguistique contrastive et traduction / Constrastive linguistics and translation*, Numéro spécial, *Le Langage et l'Homme* 34-1, Leuven, Peeters.

Čermák, F. (2002): "Today's Corpus Linguistics. Some Open Questions," *International Journal of Corpus Linguistics* 7-2, p. 265-282.

Friedl, J.E.F. (2002): *Mastering Regular Expressions*, Sebastopol (USA), O'Reilly.

Papineni, K., Roukos, S.,Ward, T. & W.-J. Zhui, (2001): "Bleu: A Method for Automatic Evaluation of Machine Translation," *IBM Research Report* RC22176, Seattle, IBM.

Sinclair, J. (1991): *Corpus, Concordance, Collocation*, Oxford, Oxford University Press.