An Automatic Reviser: The TransCheck System

Jean-Marc Jutras
RALI, Université de Montréal
C.P. 6128, succ. Centre-ville, Montréal (QC), Canada
jutras@iro.umontreal.ca

Abstract

Over the past decade or so, a lot of work in computational linguistics has been directed at finding ways to exploit the ever increasing volume of electronic bilingual corpora. These efforts have allowed for substantial expansion of the computational toolbox. We describe a system, TransCheck, which makes intensive use of these new tools in order to detect potential translation errors in preliminary or non-revised translations.

Introduction

For the sake of argument, let's consider a translator to be a black box with source text in and target text out. We feed that box with texts and, to be really tricky, we input the same text a couple of times. Looking at the results, the first thing we notice is that though the different translations are quite similar, they're not exactly the same. Nothing to worry about, this may simply exemplify the potential for synonymy and paraphrase. But let's further suppose the text to translate is too big for one individual to translate in the given time frame. In realistic conditions, such a text would be split among perhaps half a dozen translators, each with his own vocabulary, experience and stylistic preferences, which would normally lead to the well known problem of non-uniformity of the translation.

It is therefore part of the normal translation process to have a reviser look at a translator's output. His job will be to spot any typos (taken in a very broad sense to include missing chapters!). Usually, at this point the translator probably has submitted the preliminary version to a spell checker, so what could be done automatically at that level has already been done. No automatic detection of typical translation mistakes has been attempted though. That's the gap TransCheck is designed to fill. The concept of a "translation checker" was initially proposed in Isabelle and al. [8] and eventually led to a demonstration prototype concerned with the detection of a very restricted type of mistake: *deceptive cognates*. In comparison, the system described in this paper goes much further toward a "real " usable translation checker by allowing for the detection of errors of omission, the comparison of diverse numerical expressions and the flagging of inconsistent terminology.

On the interface side, it allows for the automatic alignment of the source and target texts, the flagging of potential mistakes and the possibility of saving any modifications made to the target text.

1 Error detection

Complete automatic modelling of the translation process is still far beyond our technical ability. The same is true of our ability to detect all types of translation mistakes. We can however, for certain well-defined sub-types of mistake, devise specific mechanisms. And if a program capable of detecting all mistakes of translation would undoubtedly be extremely useful, so would

one capable of detecting frequent mistakes, especially when time is short and a thorough revision isn't possible. Errors are then bound to escape the reviser's attention from time to time. This will not necessary be the case of an "automatic reviser", though. In that respect, we can compare TransCheck's behaviour to the familiar "find and replace" now common to every text editors. Who would know consider doing that particular task by hand? We now give a short description of those sub-problems TransCheck is addressing.

1.1 Errors of omission

The ability to automatically detect unintended errors of omission would be much valued, as they can prove quite embarrassing to the translator. Yet a diversity of situations can lead to such errors among which translator's fatigue and the accidental pressing of a key in a text editor, as was pointed out by Melamed [12]. Unfortunately, detecting an omission is far from being simple when taken in all its generality (from omission of single words to whole chapters). This is due in part to the fact that one language may express some ideas with a greater economy of means than another, so length difference alone isn't sufficient to identify omitted text. Consider:

- French: Quant à la section 5, elle fournit les résultats de nos simulations, que suit notre conclusion, à la sixième et dernière section.
- English: Section 5 describes our simulation results and the final section concludes.

Excluding punctuation, the French sentence in the example above has twice as many words as its English counterpart. Yet there's nothing wrong with the French translation. The task is therefore to determine whether or not correspondence at the word level is scattered throughout the whole aligned segment. Word alignment in general tends to be rather fuzzy though, as the following example shows:

- French: Voici le plan du document.
  Literal translation: (Here's) (the) (plan) (of the) (document)
- English: The paper is organized as follows.
  Literal translation: (Le) (papier) (est) (organisé) (comme) (suit)

Independently of the exact method used, alignment at the word level for this pair of sentences would prove to be rather weak. It should be noted however that the above examples are extreme cases and, without being extremely rare, they aren't exactly typical either. They're still a reminder that small omissions are unlikely to be detected with sufficient precision considering the methods available to TransCheck.

1.2 Normative usage of words

Entire repositories of usage mistakes and other linguistic difficulties of translation from English to French have been written to help language professionals become aware of them (Colpron [3], Dagenais [5], De Villiers [6], Rey [14], Van Roey and al. [17]). Unfortunately, those books are only useful to confirm existing suspicions. To warn the unsuspecting translator, TransCheck incorporates a repository of that nature.

What's particular about some of these words, and of interest for an *automatic reviser*, is that they cannot be detected by a simple dictionary lookup, for they do appear in a monolingual dictionary. What's wrong isn't the words themselves but the context in which they are used. Consider, for example, the English word *definitely* (en effet) together with the French *définitivement* (for good, once and for all). Though very similar in form, and both acceptable adverbs in their respective languages, they simply do not mean the same thing. TransCheck, therefore, looks through aligned pairs of sentences for such forbidden word pairs. It also looks for other types of mistakes, for example calques, which could potentially be detected

by a complex dictionary lookup. Calques consist of sequences of legitimate words that incorrectly mimic the structure of the other language by being sort of literal translations.

1.3 Numerical expressions

A variety of phenomena can be found under this heading (telephone numbers, percentages, fractions, etc.). One important point these otherwise very diverse types of constructions have in common is that, being open sets, they cannot be listed in repositories. Therefore, their detection will require the use of grammatical tools of some sort. But identification is not enough in most cases. Having simply identified "2" in one text and "two" in the other will not alone permit their comparison. Conversion toward a common form is required. Part of this normalised form must also indicate the type of phenomenon observed. This is so because, though there is a *6* underlying the ordinal *sixth*, only alignment with an other ordinal of the same value could be considered an appropriate match. In TransCheck, recognition, normalisation and phenomenon identification of numerical expressions are done through appropriate transducers as will be shown in the next section.

1.4 Terminological coherence

It's not rare for two or more terms to refer to the same concept. However, all things being equal, it's generally taken to be bad practice to use more than one of the synonyms for technical terms in a given translation. Failure to follow this is referred to as terminological inconsistency. To try and minimise this problem, each translator working on a project is given specific instructions that involve standardising terminology. Unfortunately, it's not rare for some translators to ignore these instructions or even for these instructions never to reach the translator. Inadequacies are therefore to be expected, and the bigger the project the more so. As an example, given the term *air bag* and possible translations *sac gonflable* and *coussin gonflable* (literally, inflatable bag/cushion), it shouldn't be allowed for both forms to appear in a given translation, though either one of the two could actually appear.

2 Tracking mistakes

We have presented briefly the type of errors detection TransCheck seeks to accomplish automatically. We will now see in more details how they are currently being implemented.

2.1 Prerequisites

In order for TransCheck to detect potential translation errors, a relatively impressive set of mechanisms is required. These include:

1. An aligner. After identification of word and sentence boundaries the text is processed into a bi-text by an alignment program. This alignment is done on the basis of both length (Gale and Church [7]) and a notion of cognateness (Simard [16]).
2. Transducers. In order to compare numerical expressions, which often diverge in format between given pairs of languages, normalisation toward a common format is required. This is done with transducers (Kaplan and Kay, [10]).
3. Part-of-speech tagger. Misleading similarities in graphical form can sometime induce translation mistakes (deceptive cognates).[1] These forbidden pairs normally involve only one of several possible parts of speech, hence the need to disambiguate them. We do this with a first-order HMM part-of-speech tagger (Merialdo [13]).

[1] In the rest of the paper, we will use *deceptive cognate* very losely often to refer to *normative usage of word* in general.

4. Translation models. Being robust, the alignment program will align a pair of texts regardless of possible omissions in the target text. To detect such omissions of text, a probabilistic bilingual dictionary is called upon. This dictionary was estimated along the line of Brown and al.'s first translation model [2]. It is used to align (coarsely) at the word level.

In what follows, we assume the reader to be at least remotely familiar with most of these mechanisms. We will however go into more technical details concerning the transducers considering the central role they play in TransCheck.

2.2 Identifying omissions

Grammatical correctors greatly relies on complex grammars to identify "typical" mistakes. We could imagine doing something similar for omission detection trying to construct the meaning of every sentences in a text and then "flag" those where semantic discontinuity were found, not unlike what a human would do. This is, of course, in our wildest dreams as, semantic analyses still remain to this day extremely elusive. Not only that, but unlike grammatical errors, we cannot anticipate something like a "typical" omission as they will appear randomly and span over any possible length of text. We must therefore recast what appears as a semantic problem in terms of more readily accessible data. The basic idea here is to assimilate an omission to a particular type of alignment where an important contiguous set of words present in the source text cannot be aligned at the word level with the target text. For this we rely on mechanisms similar to those described in Russell [15].

We can distinguish between small (a couple of sentences) and big omissions (any thing bigger than a few paragraphs). One might expect the detection of whole missing pages and chapters not to be difficult, but that's not necessarily true, as the burden of the problem then falls on the aligning program instead of the checker *per se*. Robustness here is the key-word since an alignment program that couldn't fall back on its feet after seeing big chunks of missing text would cause TransCheck to output only noise thereafter. The alignment program we use is one such robust program which, as a first step, seeks to approximate the real alignment by drawing lines in regions with high densities of cognate words. Since the distribution of cognates is *a priori* uniform throughout the text, omitted sections, when big enough, will show up on the appropriate graph as an important discontinuity in those approximation lines. As the omissions become smaller and smaller, however, the cognate's uniform distribution hypothesis becomes increasingly questionable.[2]

Still, we are interested in detecting missing sentences with acceptable precision. Ideally, this should be reflected as an X to zero alignment, but alignment programs tend to associate a high penalty to these cases, preferring to distribute extra text on adjacent regions. In order to recover from these mergings, TransCheck takes a closer look at pairs of aligned texts whenever the length ratio between source and target text falls under a certain threshold. It then attempts to aligned those pairs at the word level using a probabilistic bilingual dictionary that was estimated on the Canadian Hansard.

The "Art" of omission detection can be seen as one of trial and error in adjusting precision and recall by choosing appropriate values for what will constitute a significant difference in length ratio, a significant span of words that can't be aligned, and the penalty to be imposed if some words

[2] The probability for there to be only a few cognates between say two paragraphs is very low for French and English, but not that low for two sentences.

accidentally align due to the imprecision of the word to word alignment algorithm.

As we have just seen, the problem of detecting a missing portion of text is, in TransCheck, closely related to that of alignment, as it can be reduced to a misalignment at the word level. All the other types of errors TransCheck is concerned with are different in that respect. Correct alignment is presupposed, and when given specific pairs of aligned "tokens" the task will be to decide whether they represent valid translations. We now present the steps involved in this evaluation.

2.3 Identification

In order for TransCheck to evaluate a translation pair, their constitutive elements must first be identified. In some cases, this process requires morphological analysis and, in other, a limited type of syntactical analysis. Both type of analysis serve, to a certain extend, a single purpose: that of expressing compactly what would otherwise be a big list of tokens (in some cases, involving numerical expressions, an infinite one). This identification step is done through appropriate transducers. Basically, there are two things to keep in mind when dealing with transducers. One is that, like finite-state-automaton, they behave like recognisers; that is, when applied to an input string, if it can parse it from start to finish, the string is accepted and otherwise rejected. The second is that when doing so, it will produce an output as a result. TransCheck relies on that last property of transducers to produce a unique representation for tokens that are different in form, but semantically identical, as we will now see.

2.4 Normalisation

Though we will normally be interested in the identification of every morphological form for a given "interesting" token, once identified, these differences will be discarded by TranCheck. Compare the examples below.

- Air bag / air bags
- $2,000,000 / two million dollars / $2 million
- June 1st , 2000 / the first of June, 2000

The examples above are all in English, but the same type of diversity can be found in French too. In Figure 1 we can see an example showing the result of both the process of identification (underlined) and normalisation (=>).

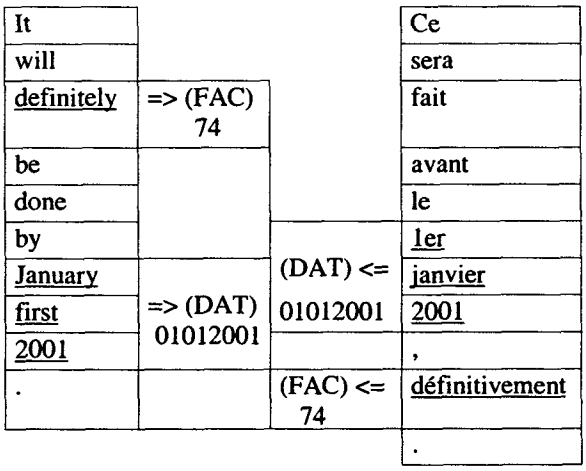

Fig. 1: Token identification and normalisation.[3]

Notice that the central part of figure 1 acts somewhat like a partial transfer[4] component (in a word to word translation model) between the French and the English texts. Though we haven't implemented it yet, this could be used to present the user with proper translation suggestions.[5]

The normalisation process depicted in figure 1, can be slightly complicated by two factors. One is the need to disambiguate the part of speech of the identified token. Consider:

---

[3] FAC stands for "faux-amis complets" (deceptive cognates in all contexts)

[4] In the case of deceptive congnates, we could talk of a forbidden transfer.

[5] Transducers can be inverted to create new transducers that will recognise what was previously outputted and output what was recognised.

- French and English: Local → **(POS)NomC**(FAC)22

Here, the *condition* field ((POS)NomC)) state that only when nouns are involved will we be in presence of deceptive cognates (but not, say, when adjectives are involved).
Consider now:

- from May 19th to 24th, 1999

Here, the dates are intermingled. The transducers we use to analyse such constructs will produced two distinct normalised forms that will both be involved in the comparison process that follows.

2.5 Comparison

The identification and normalisation process described in the previous two sections are common to deceptive cognates, technical terms and numerical expressions altogether. However, the comparison of the resulting normalised forms as well as the processing they should further undergo is of a rather case specific nature.

During the comparison process, TransCheck will only be concerned with the normalised forms resulting from the previous transduction process (the two central columns in figure 1). Each of these two columns will be considered as a set in the mathematical sense. As a consequence, the English sentence in figure 1 and the one given below are indistinguishable from TransCheck's point of view.

- It will definitely, and I mean definitely, be done by January first, 2001.

Of course, both occurrences of the word *definitely* will be flagged if the decision to flag either one is eventually taken. Each of these two sets will then be split into up to three subsets depending on whether they correspond to numerical expressions, deceptive cognates or technical terms. At this point the comparison process will be very simple. Given these subsets, the matching conditions will simply amount to the following:

- If a numeral expression appears in one language but not in the other, flag it.
- If a deceptive cognate appears in both languages, flag it.
- If a term was requested to be flagged, flag it.

2.6 Putting it all together

To recapitulate, the transducers we use in TransCheck all have the general form:

***String of interest* →**
(condition)(type)identifier

If a transducer identifies a *string of interest* and if boundary conditions are met, information about the nature of the string will be outputted. In a second step, the information from one language will have to be matched against the information from the other in accordance with the *condition* imposed by the specific nature of the identified strings.

3 The TransCheck Prototype

In the previous section, we have described what happens when a bi-text is submitted to TransCheck. We now turn to the steps that will lead to a request.

Currently, TransCheck's interface is implemented in Tcl/Tk. This has allowed us to develop a *proof of concept* without preoccupying ourselves with word processing particularities. The down side to this is a limitation to ascii characters that will eventually have to be overcome by making TransCheck part of a text editor not unlike a spell checker.

But for the time being, a TransCheck session would look something like this: The user selects through an interface a French and an English text specifying with a radio-button which of the two is the source text.[6]

[6] The system was initialy developed having in mind

Then the name of an alignment file is supplied (it will be created if it doesn't already exist). These are the minimal steps that must be taken before any analysis can take place. If, at this point, the bi-text is submitted for analysis, TransCheck will use all of it's default values and, after some window pop-up and progress report, a window containing the target text will appear on screen together with the source text facing it. All the potential errors will appear highlighted. At this point, the user can modify the target text to correct any found errors. When the session ends, the modified text will be saved (together with the appropriately modified alignment file).

We've just seen TransCheck's default behaviour. The user is also offered some customisation possibilities. This includes highlighting only those type of errors of interest to the user and setting the alignment parameters. The omission detection parameters can also be modified through an interface. Also, since as with any normative judgement, what is and what isn't a "correct" form will always be subject to debate, TransCheck allows the user to silence those alleged mistakes causing too much noise on a given text. Finally, the human reviser is allowed, any time during a session, to modify TransCheck's behaviour so that newly identified incorrect terms will be flagged thereafter, this to ensure that none of subsequent occurrences of these errors will escape his attention. This list of forbidden terms can be saved in order to constitute client specific databases so that identified problems will not be lost between projects.

4 Further development and discussion

At present, TransCheck allows for only limited customisation. However, we are well aware that the repositories available for say deceptive cognates are costly to develop and tend to include only those mistakes having a certain "history" (stability over time). That suggests the user should be allowed to add new pairs of prohibited translations on the fly. In most cases, however, adding new behaviour is a complex process available only to the system's designer because of morphology and part-of-speech considerations. Added flexibility in this regard seems mandatory. Since we cannot expect the human reviser to concern himself with such technical details, these would have to be hidden from him through adequate input interfaces. This flexibility seems to be desired independently from the now emerging problem of *localisation.*[7] We are currently addressing these issues one at a time.

So far, we have described the types of errors TransCheck is concerned with, the way they are handled and how some aspects of the processing can be customised. No figures as to precision and recall have been given though. This is in part due to the difficulty of finding preliminary translations and in part to TransCheck's customisability. For example, performance on omission detection will ultimately depend on the user's selected values. It seems to us that the best way to address both of these problems should be to actually put the system in the hands of human revisers and monitor the changes they would actually choose to make. Efforts in that direction are currently being made.

Conclusion

To our knowledge, TransCheck is still unique among *text checkers* in addressing the problem of translation errors. For a long time, only a concept without form, TransCheck, as presented in this paper, has shown the concept of a *translation checker*

English as the source text. Currently, this is still reflected only in the deceptive cognate database.

[7] Adaptation of a text for use in a different region. For example, Canadian postal code (A1B 2C3) compared to American Zip Code (12345).

to be sound and realistic. Admittedly, a lot of work, especially on the specific grammars, still has to be done. But all this now seems like a worthwhile effort considering that the resulting program could help translators considerably in their efforts to meet the quality requirements and tight deadlines they are frequently facing. We have also stressed TransCheck's adaptability to be somewhat limited. The problem seems more one of ergonomics than of principle, though. Interfaces would have to be devised to guide users through the sometime complicated steps associated with adding new restrictions. We are now considering the possibility of integrating TransCheck in an off-the-shelf text editor to cross the *ascii barrier.*

Acknowledgements

I would like to thank Elliott Macklovich, Claude Bédard, Michèle Lamarche and Guy Lapalme for their invaluable comments on drafts of this paper.

References

[1] Brown P., Cocke J., Della Pietra S., Della Pietra V., Jelinek F., Lafferty J., Mercer R., Roosin P., A. (1990) *Statistical Approach to Machine Translation.* Computational Linguistics, 16, pp. 79-85.

[2] Brown P., Della Pietra S., Della Pietra V., Mercer R. (1993) *The Mathematics of Machine Translation: Parameter Estimation tatistical Approach to Machine Translation.* Computational Linguistics, 19, pp. 263-311.

[3] Colpron, G. (1982) *Dictionnaire d'anglicismes.* Laval (Québec), Éditions Beauchemin.

[4] Dagan, I and Church K. (1997) *Termight: Coordinating Humans and Machines in Bilingual Terminology Acquisition.* Machine Translation, 12, pp. 89-107.

[5] Dagenais, G. (1984) *Dictionnaire des difficultés de la langue française au Canada.* Boucherville, Les Éditions françaises.

[6] De Villiers, J.-É. (1988) *Multidictionnaire des difficultés de la langue française.* Montréal, Éditions Québec/Amérique.

[7] Gale, W., Church K. (1991) *A Program for Aligning Sentences in Bilingual Corpora.* Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, pp. 177-184.

[8] Isabelle P. and al. (1993) *Translation Analysis and Translation Automation.* Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-93), Kyoto, pp. 201-217.

[9] Justeson, J. and Slava K. (1995) *Technical Terminology: some linguistic properties and an algorithm for identification in text.* Natural Language Engineering, 1, pp. 9-28.

[10] Kaplan, R. M., Kay, M. (1994) *Regular Models of Phonological Rule Systems,* Computational Linguistics, 20, pp. 331-378.

[11] Macklovitch, E. (1996) *Peut-on vérifier automatiquement la cohérence terminologique?* Meta, 41, pp. 299-316.

[12] Melamed, I. D. (1996) *Automatic Detection of Omissions in Translations.* In the 16th International Conference on Computational Linguistics. Copenhagen, pp. 764-769.

[13] Merialdo, B. (1994) *Tagging English Text with a Probabilistic Model.* Computational Linguistics, 20, pp. 155-168.

[14] Rey J. (1984) *Dictionnaire sélectif et commenté des difficultés de la version anglaise.* Paris, Éditions Ophrys.

[15] Russell, G. (1999) *Errors of omission in translation.* Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-99), Chester, 1999, pp. 128-138.

[16] Simard M., Foster G. and Isabelle P. (1992) *Using Cognates to Align Sentences in Parallel Corpora.* Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-92), Montréal, pp. 67-81.

[17] Van Roey, J., Granger S. and Swallow J. (1988) *Dictionnaire des faux amis français-anglais.* Paris, Duculot.