SEVENTH FRAMEWORK PROGRAMME

THEME ICT-2011.4.2(a)

Language Technologies

ACCEPT
Automated Community Content Editing PorTal

www.accept-project.eu

Starting date of the project: 1 January 2012

Overall duration of the project: 36 months

Definition of Pre-Editing Rules for English and French

Workpackage n° 2	Name: Definition of Pre- Editing and Post- Editing rules
Deliverable n° 2.1	Name: Definition of Pre-Editing Rules for English and French
Due date: 31 December 2012	Submission date: 21 December 2012

Dissemination level: PU

Organisation name of lead contractor for this deliverable: Acrolinx

The research leading to these results has received funding from the European Community's Seventh Framework Programme (*FP7/2007-2013*) under *grant agreement n° 288769.*

Contents

Definition of Pre-Editing Rules for English and French

Objectives of the Deliverable

The first half of WP2 of the ACCEPT project covers the development of pre-editing rules for both English and French. This deliverable describes the different methods we have used for the development of pre-editing rules, and the expected requirements we have identified for these rules in the ACCEPT context. Furthermore, the deliverable describes the methodology for evaluating the rules, the different evaluation options, as well as the plan and timetable for the evaluation tasks to be refined and executed in WP9.

The methods, tasks, and plans presented here are partly derived from previous work, experiments and experiences in WP2 in projects months 1 to 12. In particular, they build on results from our publications (Rayner et al. 2012) and (Roturier et al. 2012). Details of this work can be found in the appendix. This deliverable focuses on the identified requirements for pre-editing rules and the evaluation methodology.

Acrolinx Rules for Pre-Editing

A pre-editing rule is a textual pattern that reformulates the source text in order to improve the translatability and output of the statistical machine translation system (SMT) being developed in WP4.

In ACCEPT, pre-editing rules are implemented in the rule formalism of the Acrolinx software. An Acrolinx rule consists of the following components (Bredenkamp et al. 2000):

- A set of linguistic patterns and exceptions that describe the problematic issue. The rule is triggered if the pattern is found in the input text.
- The locality of the issue, that is, the problematic words.
- A list of suggestions which, if used as a replacement for the problematic words, may fix the issue. Some rules may not have a suggestion.
- A help text that provides further textual information on the issue to the user.

In the standard Acrolinx workflow, the problematic issues in the text are automatically highlighted according to the rules and presented to the user. It is then up to the user to improve the text by reading the help text, choosing a replacement suggestion, manually changing the text or ignoring the marking. The goal is to improve the quality of the given text in terms of spelling, grammar, style, terminology, and overall readability.

In contrast to requirements for the standard Acrolinx workflow, we found that the pre-editing context imposes substantially different requirements on the rules. Here, the goal is to change the text in such a way that the SMT translation output (but not necessarily the input) is improved; the process should also be as automatic as possible. The following section describes these requirements in more detail.

Main Requirements

The most important requirement for a pre-editing rule is that it should find issues that negatively affect the SMT translation output, and if possible provide a correction option that improves the translation output. It is not required that a pre-editing rule also improves the input text, but this makes it easier to present the rule to the user. Also, depending on the targeted community, rules should be automatically applicable.

These three factors – improving SMT output, presentation to the user, automatic application– impose novel requirements on the development of Acrolinx rules, and are described in more detail in the following sections.

SMT Output Improvement

In the ACCEPT project, the following language pairs are considered: English-French, English-German, English-Japanese, and French-English. Pre-editing rules are thus being developed for the source languages English and French. Since the effect of English pre-editing may differ for French, German and Japanese translation outputs, English pre-editing rules for one target language will not necessarily be good for another.

The development of pre-editing rules in ACCEPT should take into account the fact that the rules need to pre-process the input for an SMT system (Koehn 2010). We have identified the following approaches to developing suitable reformulations:

1. Words and phrases that do not appear in the training corpus of the SMT system lead to inferior translation results: we can aim for rules that replace such words and phrases by semantically equivalent substitutes.
2. Since the SMT training corpus is generally free of spelling and grammar mistakes, it is a valid assumption that identifying disfluency issues in the input data and fixing them helps the translation system.
3. There is already a large set of pre-existing rules developed by Acrolinx, among them many rules that had been previously identified as suitable for pre-editing for rule-based machine translation system. We can examine how these pre-existing rules affect the translation quality when applied automatically to the input text.
4. The SMT baseline system under consideration was trained with Symantec product manuals and other text that is mostly formal register. In contrast, the input data (forum posts) are usually written in an informal way. Pre-editing rules that address this register mismatch can therefore also help the translation system.

We have followed all four approaches in WP2, and described two of them in our publications (Rayner et al. 2012) and (Roturier et al. 2012). Details and results of this work can be found in the appendix.

Input Text Improvement

Among the pre-editing rules that fulfil the main requirement, we can distinguish between those that improve the quality of the input text in the source language and those that do not. This distinction has implications for the way in which the rules are presented to users, and how the rule results can be used.

In many cases, we found that rules which improve the input text quality on a lexical level also improve the translation quality. Examples for such lexical issues are spelling mistakes,

homophone/tense confusion, informal language, etc. The reason is that the SMT training corpus is more formal and generally contains fewer such issues. Correcting such issues thus results in a higher probability of generating matches in the translation model and leads to better translations. An advantage is that such rules are not only useful when preparing the text for MT processing, but also provide a way to improve the source language content directly.

However, it is not true in general that improvement of the input text results in a better translation. In (Roturier et al. 2012), we found that many pre-existing Acrolinx style rules in particular do not have a reliably good effect.

Conversely, some rules degrade the quality of the input text, but actually improve the quality of the translation. There are reformulations that clearly have a good effect on the translation due to the content and domain of the MT training corpus. Examples are:

- Changing the word order to match the target language, thereby making the text. ungrammatical in the source language, for example moving the French clitic “le” after the verb and replacing it by “ça” (changing *je le compile* to *je compile ça*).
- Replacing words by synonyms, thereby making the input not obviously better, for example replacing “salut” by “bonjour”.
- Changing the semantics slightly, like replacing French informal “tu” by formal “vous”, which is more frequent in the training data.

As reformulations like these do not improve the input text, they should not be shown to the users, since that could easily confuse them. Instead, these reformulations are passed directly to the MT system. We therefore need to distinguish them clearly from ones that do improve the input text.

Manual and Automatic Application

Pre-editing rules can also be differentiated along a second dimension. Some rules require human intervention (by means of the ACCEPT pre-editing plug-in), and others offer a correction that can be automatically applied.

Whether rules should be tailored towards automatic application depends on the type of user involved. Useful guidelines can be derived from post-editing evaluations that have already taken place in other Work Packages. WP7 has shown that Symantec forum users are very unlikely to edit their content once the editing task gets too complicated. In contrast, evaluations with Translators Without Borders (TSF) in WP8 have shown that all the professional translators who post-edited content using ACCEPT enjoyed the task, and may thus also be more open to pre-editing text.

For the forum user base, a requirement is therefore that rules should as far as possible be automatically applicable. To be automatically applicable, a rule should:

- not produce false positives
- be automatically correctable without user interaction required.

This means the rule has to have a high precision, and it must provide a unique (and correct) replacement suggestion. We found this to be a novel challenge for the development of rules, since typical Acrolinx rules have a different target audience: professional authors who prefer flag-

supported manual editing over full automation, and do not necessarily rely on relevant, high-precision suggestions.

There may be pre-editing rules that generally identify sections that are hard to translate, but which cannot be reliably applied in an automatic way. Such rules need to be presented to the user, and should therefore meet the following requirements:

- Improve the input text. The goal "improving for MT input" is a more abstract goal. It may not be a convincing argument for many users, in particular if the rule alters the text in a way that leaves the quality unchanged or even degrades (see above).
- The rule should have no more than three replacement suggestions to choose from, with at least one correct suggestion among them. Examples are rules for agreement errors, which often provide two grammatically correct suggestions, of which only one will normally fit the semantics of the given context.
- If the rule cannot provide a suggestion, the pre-editing task should not be overly broad, like "make sentence shorter".
- Provide simple help texts that describe the found issue and why it should be fixed.

Evaluation Methodology

In WP9, the impact of the developed pre-editing rules is measured. There are two general evaluation methods:

- **Manual evaluation:** human evaluators judge the translation output of pre-edited and unmodified source sentences.
- **Automatic evaluation**: automatic metrics are used to measure the similarity of MT outputs to given reference translations.

For pre-editing rules that need human interaction, another task is required before that:

- Manual pre-editing of input text according to pre-editing rules.

During the manual pre-editing task, we will also evaluate the usability of the pre-editing rules. The following sections explain the two evaluation forms and the pre-editing task in more detail.

Manual Evaluation Setup

For manual evaluation of translation output, we set up a contrastive evaluation task where the translations of the pre-edited and unmodified sentences are shown in random order, with the differences between the sentences marked in red. We set up both monolingual and bilingual tasks, which means the (modified) source language sentence is either shown or not shown to the evaluator. We also evaluate rules independently and in sequence. The evaluator has the task of ranking the two translations against each other according to the following scale:

- first translation is clearly better
- first translation is slightly better
- about the same
- second translation is slightly better
- second translation is clearly better

For the evaluation of individual rules, the results of the evaluation are grouped by the pre-editing rule. For evaluations of optimal sequences in which the rules are applied, we also take into account this sequence. We count how often a rule has an improving effect, and how often it has a degrading effect. We also take into account the differentiation “clearly better” and “slightly better”, and calculate the significance of the results.

Automatic Evaluation Setup

For automatic evaluation, we use the following automatic metrics: BLEU, GTM, Meteor, and TER. We summarize our experience with these metrics in (Roturier et al. 2012). Looking ahead, we plan to develop a new automatic and task-oriented metric that takes into account the special ACCEPT evaluation context involving user-generated content, the experimental setting, and additional information available elsewhere in the project. All of these metrics are described in more detail in deliverable D9.1. Since they require a reference translation, they can only be used for test sets where a reference translation is provided.

We also plan to investigate Quality Estimation techniques that do not rely on a reference translation, but instead take different indicators for the translation quality into account. Details of these indicators can be found in deliverable D 9.1.

Pre-Editing Task

For pre-editing rules that require human intervention, we first set up a manual pre-editing task. Users are given the task of pre-editing their text based on the results of rules that cannot be applied automatically. The impact of the obtained data and thus the suitability of the rules for pre-editing is then evaluated using the evaluations described above.

To get more representative results, we also aim to rate the usability of these rules, similarly to the studies of post-editing rules carried out in WP7. In a post-task questionnaire, users are asked whether they agree or disagree with the following statements:

- Pre-editing using the tool was easy.
- Pre-editing was quick.
- I did not encounter technical issues.
- I understood all the functionalities.

The data can be collected either by setting up a dedicated experiment, or by collecting feedback from community members who create new content and make use of the ACCEPT plug-in for pre-editing.

Evaluation Conditions

There are a number of dimensions along which evaluation tasks can be carried out. This section describes the different possible data sources, language pairs, and evaluation tools.

Data Sources

Two principle data sets are available for evaluation work:

1. Existing test data provided by Symantec and TWB, including reference translations required for evaluation with the automatic metrics.

2. New content produced by the Symantec and TWB communities, for which the pre-editing task described in the previous section can be directly integrated into the community software.

For each of these sets, we can either apply rules automatically (for rules that allow this), or set up the manual pre-editing task (for rules that require manual pre-editing).

Language Pairs

For the French-English language pair, the evaluation task consists of rating the French pre-editing rules according to the English MT output. For the English-German and English-French language pairs, the evaluation task consists of ranking the English pre-editing rules against German and French MT output. We do not plan to measure the impact of English pre-editing rules on Japanese in ACCEPT. However, we do consider Japanese post-editing rules and evaluation tasks once post-editing rule development starts in project month 18.

Evaluation Software and Tools

We plan to use the following software and platforms:

- **Moses SMT system:** The baseline system for Symantec and TWB as prepared in WP4.
- **Amazon Mechanical Turk:** A web-based evaluation platform that can be used to distribute evaluation tasks to users around the world. We have already set up a ranking task for French pre-editing rules, and have successfully used the platform in the experiments described in (Rayner et al. 2012).
- **ACCEPT plugin for pre-editing:** Used for the manual pre-editing task.
- **AutoApply client:** An Acrolinx client that checks a text with Acrolinx, retrieves the result, and automatically replaces marked regions by their suggestions.
- **Automatic metrics:** Implementations of BLEU, TER, Meteor, and GTM that rate MT output against a given reference translation.
- **ACCEPT Evaluation API:** The API can be used to aggregate and display the results of an evaluation or questionnaire. While the API does not provide a user interface of its own, it can be used to combine the evaluation data from a number of sources.

To combine and integrate these tools, we have already developed the following approaches:

- **extractReports:** a tool to extract flag information from Acrolinx check reports, to automatically apply suggestions, to automatically translate the suggestions and to prepare manual MT output evaluation tasks.
- **Rule scoring framework:** a framework to automatically apply suggestions, automatically translate the suggestions, and automatically rank the translations.

Evaluation Tasks

Summarizing the previous section, we get a large number of evaluation options:

- Symantec or TWB data
- derived from provided test set or from “real” community data
- automatically or manually pre-edited

- rules in sequences / not in sequence
- English or French sources, with French, German or English target sentences
- evaluation using Amazon Mechanical Turk or via automatic metrics if reference translations are provided.

Evaluation Plan and Timetable

To make the evaluation task tractable, we will settle on specific combinations of the various evaluation options described above and carry out evaluations in WP9 along the following plan for the project months 13 to 18:

M13: Evaluation Tool Setup

- identify to what extent the rule scoring scripts and the extractReports tool can be merged to provide a more unified evaluation framework
- change tools to aggregate and display results via the ACCEPT Evaluation API

M14-16: Automatic and Manual Evaluation of Rules for Symantec and TWB Content

This evaluation directly builds on methods and experiments previously developed in WP2 (Roturier et al., 2012; Rayner et al. 2012)

- Data: Symantec and TWB test data with reference translations
- English-French, French-English language pairs
- automatic application of rules
- evaluation of rules using automatic metrics and Amazon Mechanical Turk evaluation tasks

M15-17: Automatic Evaluation for User-Generated Symantec Content

This evaluation includes different language pairs. If quality estimation features can be identified in WP9, this evaluation should also include these features.

- Data: content generated by users in Symantec community
- English-French, English-German, French-English language pairs
- automatic application of rules
- evaluation using automatic metrics and possibly quality estimation techniques

M16-18: Manual Evaluation of TWB Content with Pre-Editing Task

This evaluation includes a pre-editing task with questionnaire.

- Data: content generated by translators in TWB community
- pre-editing tasks with questionnaire
- English-French, and French-English language pairs
- Amazon Mechanical Turk evaluation task

Bibliography

Bredenkamp, A., Crysmann, B., and Petrea, M. (2000). Looking for Errors: A Declarative Formalism for Resource-Adaptive Language Checking. Proceedings of *LREC 2000*. Athens, Greece.

Koehn, P. (2010). Statistical Machine Translation. Cambridge University Press.

Rayner, M., Bouillon, P., and Haddow, B. (2012): Using Source-Language Transformations to Address Register Mismatches in SMT. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, October 2012, San Diego, USA.

Roturier, J., and Bensadoun, A. (2011). Evaluation of MT Systems to Translate User Generated Content. In *Proceedings of MT Summit XIII: the Thirteenth Machine Translation Summit*, Xiamen, China.

Roturier, J., Mitchell, L., Grabowski, R., and Siegel, M. (2012). Using automatic machine translation metrics to analyze the impact of source reformulations. In *Proceedings of AMTA*, San Diego, CA, USA.

Appendix: Previous Work on Rule Development

As described in the deliverable, we have identified a rule development methodology consisting of four parallel approaches:

1. **Words and phrases that do not appear in the training corpus of the SMT system lead to inferior translation results: we aim for rules that replace words and phrases of this kind by semantically equivalent substitutes.**

Following this approach, we conducted an experiment for French pre-editing rules where we used frequencies of bigrams and trigrams to find short phrases common in the Symantec forum data, but rare in the SMT training corpus. We ordered these bigrams and trigrams by decreasing frequency, and extracted sentences that contained them. With this method, we could speed up the manual identification of inferior English translations. We were also able to replace the problematic French word patterns by semantically similar phrases that produced much better translation results. From these replacements, we could derive automatic pre-editing rules. Examples include:

- replace *il faut que vous* with *vous devez*
- replace *je confirme/ …avoir* with *je confirme/ … que j'ai*
- replace *autant pour moi* with *je suis aussi désolé*

2. **Since the SMT training corpus is generally free of spelling and grammar mistakes, it is a valid assumption that identifying typical issues in the input data and fixing them helps the translation system.**

We identified typical issues in the forum data that consistently produce inferior translation output. For French, they include among others:

- homophone confusion: *a/à, qu'elle/quelle, si/ci, ou/où, ce/se, ma, mi/m'a, m'y, la/là, sur/sûr, des/dès, tous/tout, quelque/quel que, du/dû*
- tense confusion such as *imperative* instead of *indicative*, *future* instead of *conditional*
- agreement errors

For English, we found other typical issues that are difficult for the SMT system to translate:

- missing apostrophes, wrong spacing: *wont*, *dont*, *some what*
- colloquial language and spelling: *wanna*
- ellipses: *hope this helps*

The findings were implemented in corresponding Acrolinx rules.

3. **There is already a large set of rules developed by Acrolinx, among them many rules that have previously been identified as suitable for pre-editing input to rule-based machine translation systems. We can thus examine how existing rules affect translation quality when applied automatically to input text.**

In (Roturier et al. 2012), we created a rule scoring framework that automatically checks the English forum data test set with existing Acrolinx rules, automatically replaces issues by suggestions, translates the text via the Symantec baseline system, and automatically evaluates the translation

output against the reference translations of the test set with the automatic metrics BLEU, GTM, and TER. In addition, we performed a manual contrastive evaluation. We counted the number of positive and negative impacts on the scores and human judgements for each rule and each target language under consideration (German and French). With this method, we were able to identify a few pre-existing rules that generally have a positive impact on the translation quality:

- noun-adjective confusion
- noun-adjective-verb confusion
- avoid contractions
- use complementizer

4. **The SMT baseline system from WP4 was trained with Symantec product manuals and other text that is mostly formal register. In contrast, the input data (forum posts) are usually written in an informal way. Pre-editing rules that address register mismatch can thus also help the translation system.**

In (Rayner et al. 2012), we observed that the available training data for the baseline system created in WP4 (product manuals and other text) is mostly formal register, while the typical input text (forum posts) contained mostly informal language. To address this register mismatch for the French input language, we created a few reformulation rules to replace informal by formal language, such as *tu* + verb by *vous* + verb, and applied them to the forum data. In parallel, we created "reversed" versions of the rules that replaced formal by informal language, applied them to the training data, and retrained the system. We set up an evaluation experiment within the Amazon Mechanical Turk platform to rank the translation outputs. Results showed that both approaches produce a significant improvement; pre-editing the input data is significantly better than producing artificial training data, and combining the two approaches is significantly better than using either one singly. This experiment also allowed us to collect valuable experience with the AMT platform and develop associated software which may well be useful in other contexts.