



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

**Monolingual and bilingual post-editing environment and
evaluation portal demonstrators**

Workpackage n° 5

Name: Portal Integration

Deliverable n° 5.8

Name: Monolingual and bilingual post-editing
environment and evaluation portal demonstrators

Due date: 31 December 2014

Submission date: 19 December 2014

Dissemination level: PU

Organisation name of lead contractor for this deliverable: SYMANTEC

Author(s): Johann Roturier, David Silva

Internal reviewer(s): Robert Grabowski

Proofreading: Manny Rayner

Copyediting: Violeta Seretan

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

| | |
|---|----|
| Contents | 3 |
| List of Figures..... | 4 |
| Foreword | 5 |
| 1 Objectives of the Deliverable | 5 |
| 2 Overview of the ACCEPT Architecture..... | 5 |
| 3 Availability of the ACCEPT Infrastructure | 6 |
| 4 The ACCEPT Post-Edit Components | 7 |
| 4.1 New Post-Edit Plug-in Functionality | 7 |
| 4.1.1 Improved Usability | 7 |
| 4.1.2 Requesting Alternative Translations for a Given Phrase..... | 8 |
| 4.1.3 Triggering Interactive Checks | 9 |
| 4.1.4 XLIFF Reports | 10 |
| 4.2 New Project Functionality | 12 |
| 5 New Portal-based Evaluation Functionality | 13 |
| 5.1 New Evaluation Project Creation | 13 |
| 5.2 New Evaluation Project User Management | 15 |
| 5.3 New Evaluation List Page | 16 |
| 5.4 New Evaluation Page | 16 |
| 5.5 New Evaluation Data | 17 |
| 6 Deployment Guidelines | 18 |
| 6.1 Possible Use Cases..... | 18 |
| 6.2 High-level Deployment Guidelines..... | 18 |
| 6.2.1 Server-Side..... | 19 |
| 6.2.2 Client-Side..... | 21 |
| 7 Conclusion | 21 |
| References..... | 21 |

List of Figures

| | |
|---|----|
| Figure 1: The ACCEPT architecture..... | 5 |
| Figure 2: Maximizing window..... | 7 |
| Figure 3: Requesting alternative translations from a paraphrasing service | 8 |
| Figure 4: Interactive checking mode | 9 |
| Figure 5: New usage notes in XLIFF report..... | 10 |
| Figure 6: Post-editing project configuration options | 12 |
| Figure 7: Internal evaluation project creation form..... | 13 |
| Figure 8: Creating evaluation questions..... | 15 |
| Figure 9: Inviting users to an evaluation project..... | 15 |
| Figure 10: Evaluation task list..... | 16 |
| Figure 11: Segment evaluation in progress..... | 16 |

Foreword

As agreed with the Project Officer on November 10th 2014, the original deliverables D5.7: *Monolingual and Bilingual Post-Editing Environment demonstrator* and D5.8: *Evaluation portal demonstrator* are being merged into the present, common deliverable (D5.8).

1 Objectives of the Deliverable

The main objective of this deliverable, which combines Tasks 5.2 and 5.3, is to describe the work that was required to turn some of the prototypes developed in Years 1 and 2 into full-fledged demonstrators. Specifically, the main goal of Task 5.2 was to enhance the post-editing client prototype by integrating the resources and technologies developed in WPs 2 and 7 (e.g., post-editing rules and request of alternative translation of phrases). The main goal of Task 5.3 was to refine the evaluation environment prototype by allowing post-edited content collected using the ACCEPT post-editing component to be evaluated using the Evaluation section of the ACCEPT Portal. The actual output of Tasks 5.2 and 5.3 is software code that has been made publicly available on the [Github.com](https://github.com) platform in the following repositories: [ACCEPT Portal](#), [ACCEPT API](#), [ACCEPT post-edit](#).¹

2 Overview of the ACCEPT Architecture

- The overall ACCEPT system can be logically described as three components: The ACCEPT API;
- The ACCEPT Web Portal;
- The ACCEPT JavaScript clients (Post-Edit, Pre-Edit, Evaluation).

The overall ACCEPT system's architecture is presented in Figure 1:

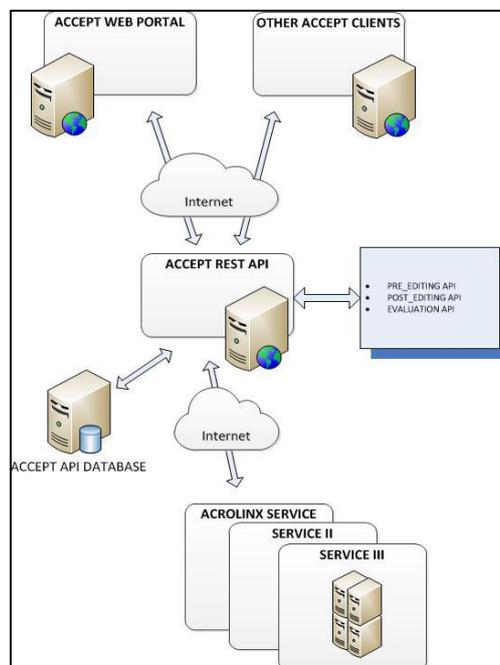


Figure 1: The ACCEPT architecture

¹ This platform replaces the code delivery platforms mentioned in previous deliverables (e.g., the ACCEPT Content Delivery Network (CDN) in [Deliverable D5.6](#)).

As shown in Figure 1, the ACCEPT architecture is extremely flexible as new back-end services – e.g., “Service II” or “Service III” – may be added to provide new functionality to the client applications (“plug-ins”). For instance, a paraphrasing service provided by the University of Edinburgh was added in Year 3 in order to provide to users of the post-editing client the ability to request alternative translations for specific phrases.² More information on this integration is provided in Section 4.1.2.

3 Availability of the ACCEPT Infrastructure

Throughout the duration of the ACCEPT project, an instance of the ACCEPT API was deployed online alongside an instance of the ACCEPT Portal (www.accept-portal.eu and www.accept-portal.com). This deployment served as a reference implementation to be used for demonstration and experimentation purposes. Going forward, entities will have the ability to deploy their own instances of the ACCEPT [API](#) and [Portal](#) since these components have been publicly released as open-source (under an [Apache v2 license](#)) on the [Github.com](https://github.com) platform, based on the following structure:

- [Repository for the ACCEPT API](#): contains two .NET projects, the ACCEPT API and the ACCEPT Framework. The ACCEPT API is a .NET MVC4 Web application embracing some of the REST standards. The ACCEPT Framework is a .NET Class Library type of project, which is responsible for implementing the ACCEPT system’s domain logic and for executing data persistence operations.
- [Repository for the ACCEPT Portal](#): contains one .NET project, the ACCEPT Web Portal. The ACCEPT Web Portal is a .NET MVC4 application, whose data is consumed from the ACCEPT API endpoint, so as part of the configuration process a valid ACCEPT API endpoint needs to be set up in its “Web.config” file.
- [Repository for the ACCEPT Pre-Edit clients](#): contains two jQuery plug-ins, the ACCEPT Pre-Edit plug-in and the ACCEPT Pre-Edit Real-time plug-in.
- [Repository for the ACCEPT Post-Edit client](#): contains one jQuery plug-ins, the ACCEPT Post-Edit plug-in. The ACCEPT Post-Edit plug-in can be used for tackling generic scenarios where content edition is required.
- [Repository for the ACCEPT Evaluation client](#): contains one jQuery plug-in, the ACCEPT Evaluation plug-in. The ACCEPT Evaluation plug-in can be used for tackling generic scenarios where content evaluation is required.³

More information on how to deploy these components is provided in Section 6.

² http://statmt.org:8123/?max=<max_results>&sys=<sys_id>&lang=<lang_code>&q=<URL_encoded_input> (Accessed: December 2014).

³ This client is based on the client described in Mitchell et al. (2014).

4 The ACCEPT Post-Edit Components

The first part of this section describes the changes that were made to the post-edit plug-in. The changes that were made to the administration of post-editing projects on the Portal itself are then briefly presented.

4.1 New Post-Edit Plug-in Functionality

The following functionality has been added in version 3 of the Post-Edit plug-in:

- Ability to maximize the main window in order to improve usability;
- Ability to request alternative translations for a given phrase;
- Ability to trigger interactive (target text) checks using a real-time approach.

While the first functionality is user-controlled, the availability of the other two is controlled by the post-editing project administrator.

4.1.1 Improved Usability

Besides minor changes to improve the overall usability of the post-editing client (e.g., changing the size of elements such as the one containing the text to edit, changing the default font sizes), an extra button was added to the User Interface (as shown in green in Figure 2) to give users the ability to maximize the post-editing window. This change was made based on received user feedback and is expected to reduce the amount of scrolling required, especially for longer texts.

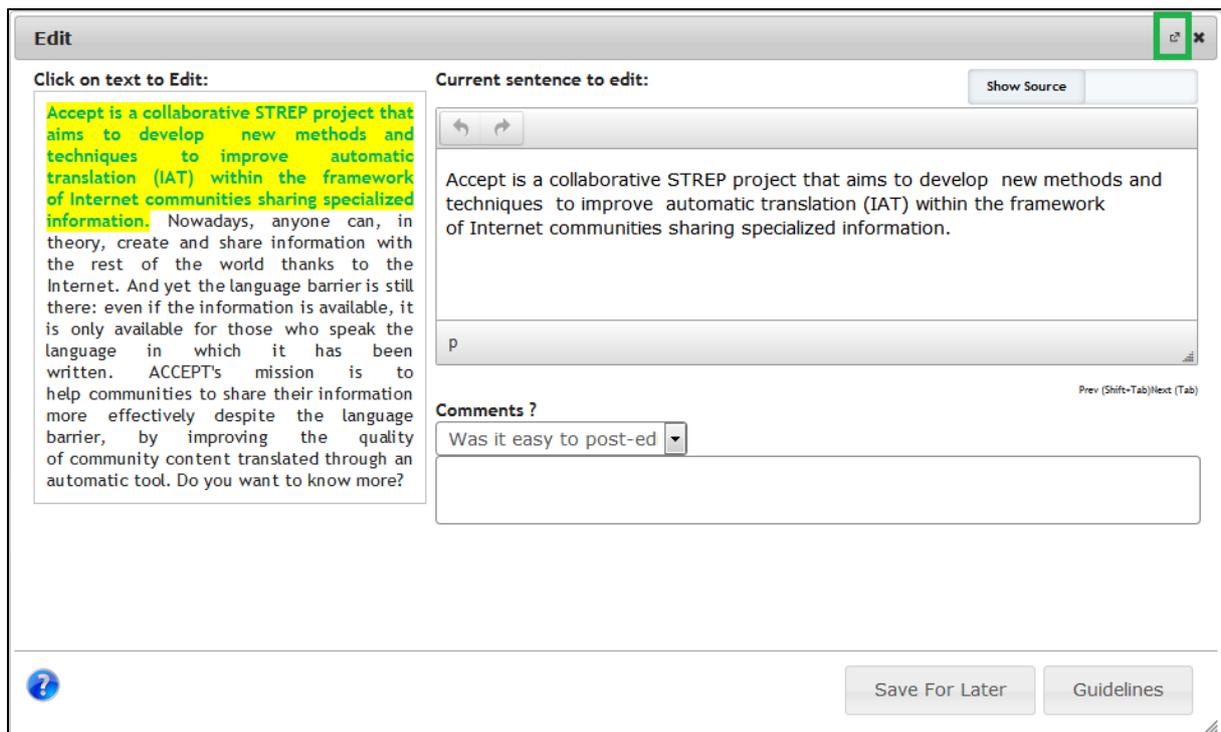


Figure 2: Maximizing window

4.1.2 Requesting Alternative Translations for a Given Phrase

When post-editing projects are configured such that alternative translations can be requested from a paraphrasing service such as the one described in [Deliverable D7.2](#), post-editors are able to highlight words or phrases that they do not understand or do not like, in order to be presented with other options. Figure 3 shows how this can be achieved using a user-defined left-click selection. If results are available from the paraphrasing service, they are displayed in a context menu. In the background, the following information is sent to the paraphrasing service:

- Paraphrasing service ID that should provide the alternative translations (e.g. “symantec”)
- Language code of the target text (e.g., “en”);
- Context of the phrase with the phrase to be reformulated marked off with " || " separators (e.g., “This is || an attempts || to check the file.”);
- The maximum number of alternative options to return (e.g., “5” to limit the results to at most 5 alternative options).

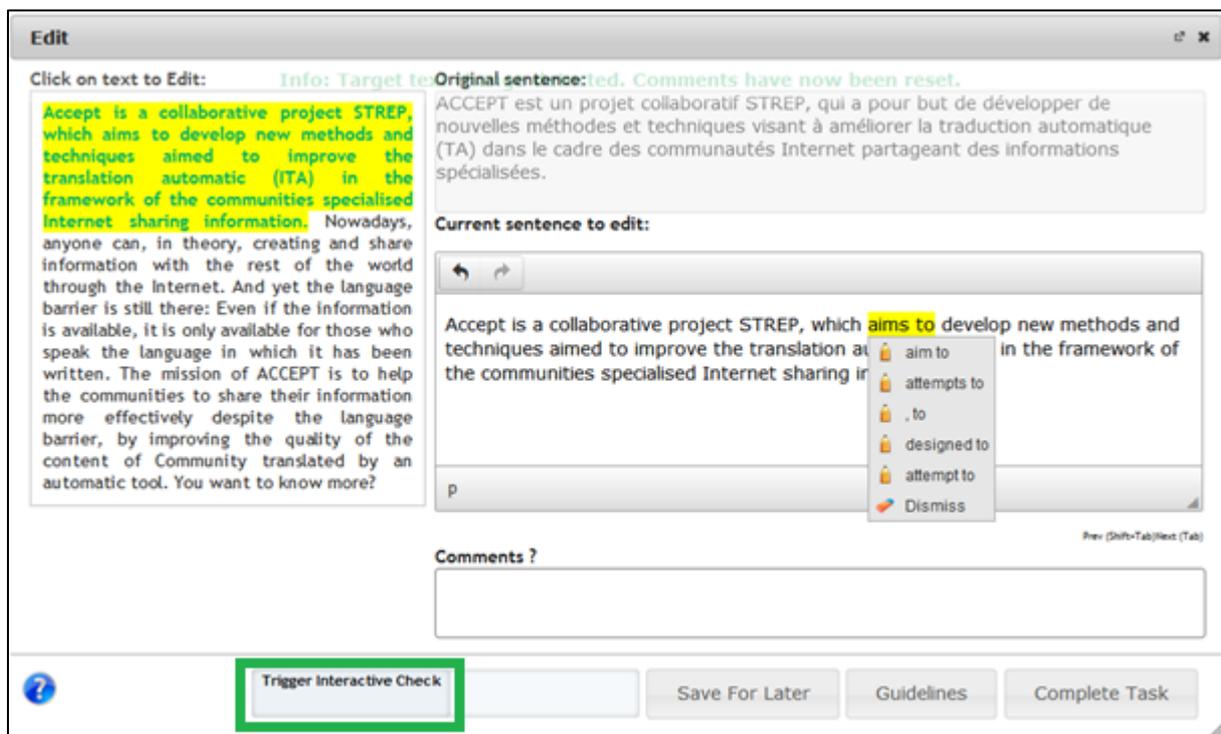


Figure 3: Requesting alternative translations from a paraphrasing service

The alternative context menu also contains a “Dismiss” option so that users have to explicitly indicate that none of the suggestions were useful (an action which is recorded in the logs). Figure 3 shows that the default post-editing environment allows users to request such alternatives unless they click the “Trigger Interactive Check” button, which brings them into the Interactive Checking mode described in the next section.

4.1.3 Triggering Interactive Checks

Version 2 of the post-editing interface allowed users to manually trigger a check for each target segment. This approach was both cumbersome (multiple clicks were required; the results were shown in a separate window) and counter-intuitive (as post-editors often check content at the document level rather than the segment-level). To address these issues, an interactive checking mode can be accessed as described in Figure 3. Once the button is clicked, each target segment is checked in an independent manner and the segment-level results are displayed both on the left side (overall target text) and right side (current segment to edit) of the post-editing interface, as shown in Figure 4. In this example, potential errors that are detected by the post-editing rule set are directly visualized in the element containing the target text to edit. This change is an important usability improvement as multiple user requests throughout the project had highlighted the need to have checking results displayed in this way, instead of using a separate pop-up window. This approach allows users to select or ignore suggestions at a segment level. Additional corrections can also be made but as soon as the user navigates away from a given segment (for instance, to focus on another segment containing problems), this newly modified segment is automatically rechecked in the background, so that users immediately know whether they may have introduced other problems while editing the text.

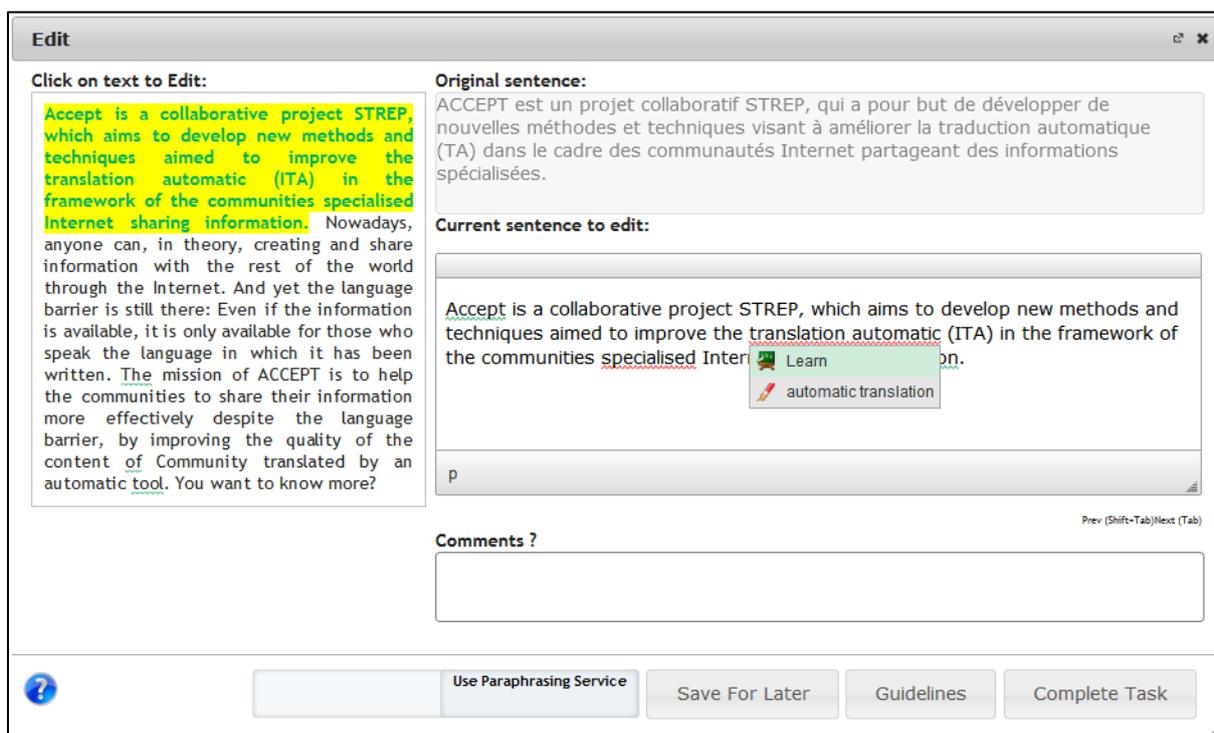


Figure 4: Interactive checking mode

The real-time check is an integrated functionality whose implementation is based on an independent jQuery plug-in. This plug-in is called the ACCEPT Pre-Edit Real-time plug-in. In practice, it is a lightweight version of the already developed Pre-Edit plug-in. Besides being more lightweight (i.e., no external dialog opens), it provides:

- A real-time experience using a [Web Sockets](#) pipeline, which means that the client no longer needs to constantly poll the server to determine whether the content is ready. The Web server will send data back to the client as soon as it is available.
- Multiple text placeholder selection through the plug-in's configuration using a server-side parallel processing mechanism. In the Post-Editing scenario, this feature is used to process all task segments in one go.

Since the ACCEPT Pre-Edit Real-time plug-in is itself a plug-in, some prior configuration is also needed. However, most of the configuration parameters are currently hardcoded in the context of a post-editing client (e.g., the placeholders for text processing are the left-hand side segments). Only two parameters require configuration:

- (string) interactiveCheckConfigPath: The configuration path allowing the ACCEPT Pre-Edit Real-time plug-in load its labels from a JSON-based configuration file.
- (string) acceptHubUrl: The endpoint from which to access the ACCEPT Pre-Edit Real-time plug-in.

4.1.4 XLIFF Reports

Special notes associated with a paraphrasing service, an interactive check or provided translation options may be included in **phase** elements, as shown in Figure 5.

```
<phase phase-name="r1.1" process-name="bilingual" date="2014-10-14T10:57:42.000Z" contact-email="">
<note annotates="target" from="paraphrasing_replacement">
2014-10-14T10:57:40.957Z||improve||enhance||99||1||2014-10-14T10:57:42.442Z||Accept is a
collaborative project STREP, which aims to develop new methods and techniques aimed to improve the
translation automatic (ITA) in the framework of the communities specialised Internet sharing
information.</note>
<note annotates="target" from="paraphrasing_display">2014-10-14T10:57:55.617Z||specialised Internet
sharing information||175||Accept is a collaborative project STREP, which aims to develop new methods
and techniques aimed to enhance the translation automatic (ITA) in the framework of the communities
specialised Internet sharing information.</note>
<note annotates="target" from="interactive_check">
2014-10-14T10:58:13.105Z||specialised||specialized||175||1||2014-10-14T10:58:17.667Z||Accept is
a collaborative project STREP, which aims to develop new methods and techniques aimed to enhance the
translation automatic (ITA) in the framework of the communities specialised Internet sharing
information.</note>
<note annotates="target" from="interactive_check">2014-10-14T10:58:18.944Z||translation
automatic||automatic translation||111||1||2014-10-14T10:58:20.274Z||Accept is a collaborative
project STREP, which aims to develop new methods and techniques aimed to enhance the translation
automatic (ITA) in the framework of the communities specialized Internet sharing information.</note>
</phase>
<phase phase-name="t1.1" date="2014-10-14T10:57:30.000Z" contact-email="">
<note annotates="target" from="paraphrasing_empty">2014-10-14T10:57:33.878Z||STREP||34||Accept is a
collaborative project STREP, which aims to develop new methods and techniques aimed to improve the
translation automatic (ITA) in the framework of the communities specialised Internet sharing
information.</note>
</phase>
</phase-group>
<count-group name="1">
<count phase-name="r1.1" count-type="x-editing-time" unit="x-seconds">39.443</count>
<count phase-name="r1.1" count-type="x-checking-usage" unit="instance">1</count>
<count phase-name="r1.1" count-type="x-paraphrasing-usage" unit="instance">2</count>
<count phase-name="t1.1" count-type="x-think-time" unit="x-seconds">11.812</count>
<count phase-name="t1.1" count-type="x-paraphrasing-usage" unit="instance">1</count>
```

Figure 5: New usage notes in XLIFF report

Such notes have a descriptive **from** attribute value, as described in Table 1.

| Value | Description | Note value format |
|--------------------------|--|--|
| paraphrasing_replacement | When a paraphrasing suggestion is used to replace a substring in the original text | paraphrasing_trigger_timestamp substring_selected paraphrasing_suggestion substring_index paraphrasing_suggestion_position paraphrasing_suggestion_use_timestamp original_text |
| paraphrasing_display | When paraphrasing suggestions are displayed and dismissed | paraphrasing_trigger_timestamp substring_selected substring_index original_text |
| paraphrasing_empty | When no paraphrasing suggestions are returned by the service | paraphrasing_trigger_timestamp substring_selected substring_index original_text |
| paraphrasing_failure | When the paraphrasing service fails | paraphrasing_trigger_timestamp substring_selected substring_index original_text |
| interactive_check | When an interactive check's suggestion replaces a substring in the original text | interactive_check_menu_display_timestamp substring_marked interactive_check_suggestion substring_index interactive_check_suggestion interactive_check_suggestion_use_timestamp original_text |
| trans_options | When a translation option is used to replace a substring in the original text | highlighted_substring translation_option_used substring_index original_text |

Table 1: Description of "from" attributes for special notes

Interactive check usage is incremented each time the user triggers a text-level check so the actual value of a phase-level **count** element, whose **count-type** attribute has an *x-checking-usage* value, may not match the number of notes with an *interactive_check* **from** value.⁴ On the other hand, paraphrasing usage is incremented each time a paraphrasing request is triggered (regardless of success and failures) so the actual value of a phase-level **count** element, whose **count-type** attribute has an *x-paraphrasing-usage* value, will match the number of notes with a **from** value starting with *paraphrasing_*. An additional phase-level **count** element, whose **count-type** attribute has an *x-paraphrasing-failures* value, may be included when failures happen (*x-paraphrasing-failures* being a subset of *x-paraphrasing-usage*).⁵

⁴ This mismatch may occur because *interactive_check* notes correspond to replacement actions, whereas *x-checking-usage* refers to the number of checks.

⁵ Failures include errors from the paraphrasing service as well as timeout errors.

4.2 New Project Functionality

The following functionality has been added in relation to post-editing project creation:

- Ability to specify a set of interactive target text checking resources (e.g., post-editing rules) for specific language pairs;
- Ability to specify a paraphrasing service for specific language pairs.⁶

The new functionality is shown in Figure 6, where project administrators are given the possibility to enable paraphrasing and/or interactive checking.

The screenshot shows a 'Create Project' form with the following fields and values:

| Field | Value |
|-------------------------------|---|
| Project Name: | |
| Organization: | |
| ProjectDomain: | ACCEPT Domain |
| Project Source Language: | English |
| Project Target Language: | French |
| UI Configuration: | Bilingual |
| Enable Paraphrasing: | Enabled |
| Paraphrasing System ID: | |
| Enable Interactive Checking: | Disabled |
| Default Rule Set: | Postediting-EN-FR |
| Enable Translation Options: | Enabled |
| Allow user to display source: | End user cannot switch Hide/Show source |

A tooltip for the 'Enable Paraphrasing' field states: 'Specify whether users should be able to trigger alternative options from a paraphrasing service.'

Figure 6: Post-editing project configuration options

For each option to be configured properly, two fields must be used. First, the value of the first field (e.g., “Enable Paraphrasing”) must be set to “Enabled” if the goal is to give post-editing the ability to use that functionality. The second field to consider is the one controlling the actual resources that will be used to generate alternative translations (i.e., a paraphrasing system ID) or editing suggestions (i.e., a post-editing rule set name). Additional system-wide paraphrasing options include the maximum number of results that will be displayed within the context menu as well as the maximum amount of time given to the paraphrasing service to return options. Both of these options were deemed necessary from a usability perspective to avoid (i) overwhelming users with too many options and (ii) frustrating users with long response times.

⁶ When this functionality is selected, the “Translation Options” functionality becomes unavailable to avoid conflicts.

5 New Portal-based Evaluation Functionality

The “Evaluate” component of the ACCEPT Portal has been updated in Year 3 to address the following shortcomings:

- Post-edited content had to be manually uploaded into an Evaluation project even when that content had been collected via the ACCEPT Post-editing API.
- Evaluation projects created in the ACCEPT Portal did not offer any default evaluation environment in the Portal. Project administrators had to create their own evaluation collection mechanism.

5.1 New Evaluation Project Creation

The Project Creation form available on the Evaluate section of the ACCEPT Portal has been enhanced to offer project administrators the ability to create “Internal” projects. While “external” projects will focus on evaluating content uploaded by an evaluation project administrator, an “internal” evaluation project gives project administrators the ability to use the content of an existing post-editing project. Figure 7 shows the additional options that are made available to evaluation project creators.

| Create Evaluation Project | |
|---|---|
| Name: | <input type="text"/> |
| Description: | <input type="text"/> |
| Organisation: | <input type="text"/> |
| API Key Domain: | <input type="text" value="www.accept-portal.eu"/> |
| Project Type: | <input type="text" value="Internal Project"/> |
| Post-Edit Project: | <input type="text" value="Select Project..."/> |
| Evaluation Approach: | <input type="text" value="Not Set"/> |
| Avoid Duplications: | <input type="text" value="Not Set"/> |
| Include Project Owner Revisions: | <input type="checkbox"/> |
| Custom Email Invitation Message: | <input type="text"/> |
| <input type="button" value="Create"/> <input type="button" value="Cancel"/> | |

Figure 7: Internal evaluation project creation form

Figure 7 shows that internal projects can be configured in the following ways by:

- Selecting a “Post-Edit Project” (also owned by the evaluation project administrator) as content source. This project does not have to be completed, which means that an evaluation project can be run concurrently with a post-editing project.

- Selecting an “Evaluation approach” among:
 - *Only the Original Document* gets selected for evaluation (i.e., the source and target content that was uploaded in JSON format into a post-editing project). Very often the target content of the original document that gets uploaded is machine-translated content.
 - *The Original Document and Last (Per Segment) Revisions* are selected for evaluation. Very often the target content of the original document that gets uploaded is machine-translated content and the last revisions (per segment) are post-edited versions of the machine-translated content.
 - *Only Last (Per Segment) Revisions* are selected for evaluation. Very often the last revisions (per segment) are post-edited versions of the machine-translated content.
- Avoiding duplications:
 - At the task level. A task corresponds to a document that was uploaded in a JSON file. Avoiding duplication at the task level means that if Post-Editor 1 and Post-Editor 2 have edited a segment in the same way in a given task, only one of these segments will be made available to a given evaluator.
 - At the project level. Avoiding duplication at the project level means that if two post-editors have edited a segment in the same way in any task of a project, only one of these segments will be made available to a given evaluator.
 - Otherwise, all segments will be made available to any evaluator.
- Including a post-editing “Project Owner’s Revisions”: by default, post-editing revisions are created for the owner of a post-editing project even though the owner may not take part in the actual post-editing process. Evaluating such revisions may therefore be superfluous.
- Adding a “custom email invitation message” to the users who will be asked to perform the actual evaluation (e.g., to provide extra guidelines, to provide a link to a training video, etc.)

More information on these configuration options can be found on the tooltips associated with the various UI labels. Once a project has been created, evaluation categories and questions should be added as already documented in [Deliverable D5.3](#). Figure 8 shows how questions can be added to a project using standard 5-point scales.

| ID | Language | Question | Action | Action Confirmation | Answers | Action | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------|----------------------|--|---|---------------------|--|--------|-------|--|----------------------|----------------------|------------------------------------|----|--------|-------|--------|----|----------|---|---|----|------|---|---|----|------------|---|---|----|-----------|---|---|----|------------------|---|---|---|
| 16 | English | How fluent is this text? | Send | Thanks | <table border="1"> <thead> <tr> <th>Answer</th> <th>Value</th> <th></th> </tr> </thead> <tbody> <tr> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="button" value="Add"/></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>ID</th> <th>Answer</th> <th>Value</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>43</td> <td>Flawless</td> <td>5</td> <td>Edit Delete</td> </tr> <tr> <td>44</td> <td>Good</td> <td>4</td> <td>Edit Delete</td> </tr> <tr> <td>45</td> <td>Non-native</td> <td>3</td> <td>Edit Delete</td> </tr> <tr> <td>46</td> <td>Disfluent</td> <td>2</td> <td>Edit Delete</td> </tr> <tr> <td>47</td> <td>Incomprehensible</td> <td>1</td> <td>Edit Delete</td> </tr> </tbody> </table> | Answer | Value | | <input type="text"/> | <input type="text"/> | <input type="button" value="Add"/> | ID | Answer | Value | Action | 43 | Flawless | 5 | Edit Delete | 44 | Good | 4 | Edit Delete | 45 | Non-native | 3 | Edit Delete | 46 | Disfluent | 2 | Edit Delete | 47 | Incomprehensible | 1 | Edit Delete | Edit Delete |
| Answer | Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="text"/> | <input type="text"/> | <input type="button" value="Add"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ID | Answer | Value | Action | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 43 | Flawless | 5 | Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 44 | Good | 4 | Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 45 | Non-native | 3 | Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 46 | Disfluent | 2 | Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 47 | Incomprehensible | 1 | Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 17 | English | How much of the source meaning is preserved in the target? | Send | Thanks | <table border="1"> <thead> <tr> <th>Answer</th> <th>Value</th> <th></th> </tr> </thead> <tbody> <tr> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="button" value="Add"/></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>ID</th> <th>Answer</th> <th>Value</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td>49</td> <td>All</td> <td>5</td> <td>Edit Delete</td> </tr> <tr> <td>50</td> <td>Most</td> <td>4</td> <td>Edit Delete</td> </tr> <tr> <td>51</td> <td>Much</td> <td>3</td> <td>Edit Delete</td> </tr> <tr> <td>52</td> <td>Little</td> <td>2</td> <td>Edit Delete</td> </tr> <tr> <td>53</td> <td>None</td> <td>1</td> <td>Edit Delete</td> </tr> </tbody> </table> | Answer | Value | | <input type="text"/> | <input type="text"/> | <input type="button" value="Add"/> | ID | Answer | Value | Action | 49 | All | 5 | Edit Delete | 50 | Most | 4 | Edit Delete | 51 | Much | 3 | Edit Delete | 52 | Little | 2 | Edit Delete | 53 | None | 1 | Edit Delete | Edit Delete |
| Answer | Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <input type="text"/> | <input type="text"/> | <input type="button" value="Add"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ID | Answer | Value | Action | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 49 | All | 5 | Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 50 | Most | 4 | Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 51 | Much | 3 | Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 52 | Little | 2 | Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 53 | None | 1 | Edit Delete | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figure 8: Creating evaluation questions

5.2 New Evaluation Project User Management

Internal evaluation projects have access to the same user management functionality that was already available for post-editing projects. Any number of users can be invited to an evaluation project by specifying their email addresses as shown in Figure 9.

Project Details

[Project](#)
[Questions](#)
[Metrics](#)
[View Data](#)
[My Project Token](#)
[Evaluate](#)
[Invite Users](#)
[<<Go back](#)

| | |
|---|----------|
| ID: | |
| Name: | |
| Organisation: | |
| Description: | |
| API Key: | 04f75491 |
| API Key Domain: | |
| Edit Delete | |

Invite user(s) to project: eval-test-mt1 ✕

Please insert the email address(es):

Note: multiple email addresses should be separated by the ";" character.

Figure 9: Inviting users to an evaluation project

5.3 New Evaluation List Page

Once a user has been invited to an evaluation project, they are presented with a list of tasks. Depending on the configuration details of an evaluation project, the task list may include different types of content to evaluate (e.g., only original content, etc.), as shown in Figure 10:

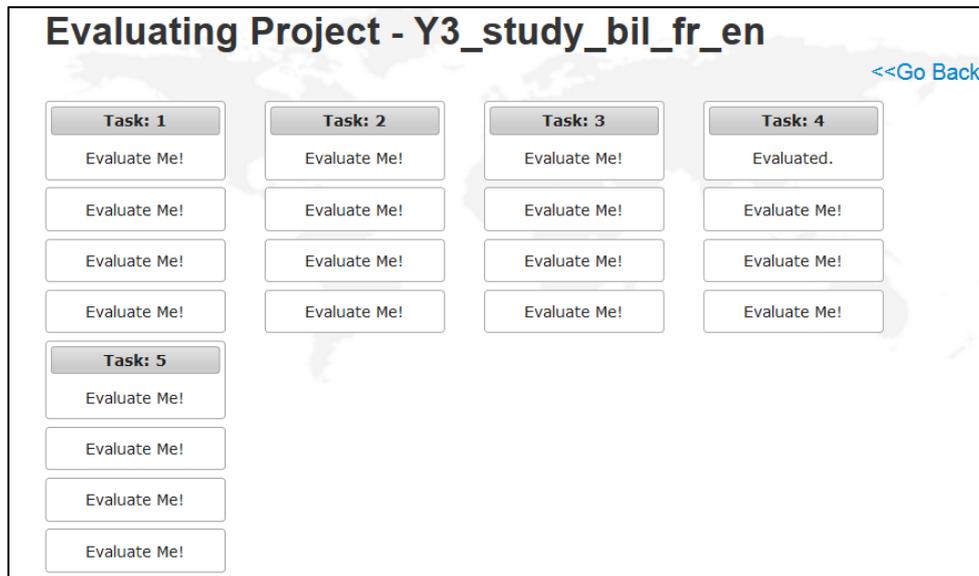


Figure 10: Evaluation task list

Evaluators are free to start evaluation tasks in any way they want by clicking one of the task’s links. Once a task does not contain any segment to rate, the task label changes (i.e., from “Evaluate Me!” to “Evaluated”).

5.4 New Evaluation Page

Each evaluation task is divided into three columns. The left column contains the source text, the middle column contains the target text (i.e., the translation, which may be a machine-translated segment or a post-edited segment) and the right column contains the question(s) that evaluators have to answer for each segment. The example in Figure 11 shows how an evaluation task progresses depending on the answers provided by the evaluator. This example contains three examples, the first of which has already been rated by the evaluator (hence being greyed out, but left on the page in order to provide context).

| | | |
|---|--|---|
| (SOLVED) I just found a setting in Group Policy to disable Geolocation. | (SOLVED) je viens de trouver un paramètre dans la politique de groupe de désactiver Geolocation. | Thanks |
| That's what I was after. | C'est ce que j'ai été après. | <p>How much of the source meaning is preserved in the target?</p> <p> <input type="radio"/> All <input type="radio"/> Most <input type="radio"/> Much <input type="radio"/> Little <input type="radio"/> None </p> <p style="text-align: right;"><input type="button" value="Send"/></p> |
| I already had the Tracking Protection lists setup. | J'ai déjà eu la protection de la liste de l'installation. | <p>How fluent is this text?</p> <p> <input type="radio"/> Flawless <input type="radio"/> Good <input type="radio"/> Non-native <input type="radio"/> Disfluent <input type="radio"/> Incomprehensible </p> <p style="text-align: right;"><input type="button" value="Send"/></p> |

Figure 11: Segment evaluation in progress

Figure 11 shows that the second segment has already been evaluated in terms of fluency, so the second question from Figure 8 (the one about adequacy) is displayed. The third segment has not been evaluated at all, so the third column displays the first question (the one about fluency). This mechanism can be extended to any question that may be present in a project.

5.5 New Evaluation Data

Internal projects collect data in a structured manner, which means that they make use of the extra parameters of the Evaluation API's **Score** method (described in [Deliverable D5.3](#)) in a specific way. When evaluation data is subsequently exported (e.g., using the Evaluation API's **Scores** method (described in [Deliverable D5.6](#)), the metadata shown in Table 2 is made available.

| Key | Value Description | Note |
|------|--|--|
| Var1 | Source segment | |
| Var2 | Target segment | |
| Var3 | Post-Edit Task ID | |
| Var4 | Username of Target segment author | This can default to "MT" if the target segment evaluated corresponds to an original segment. |
| Var5 | Mapping hash consisting of: source segment + target segment + projectToken + textIdInEvaluation + userInEvaluation | This hash may be useful to determine whether a specific segment has already been evaluated by a given evaluator. |
| Var6 | Duplication control hash | If the duplication is set at the project-level: source segment + target segment + textIdInEvaluation + projectToken + userEvaluating + uniqueQuestionControlHash) If the duplication is set at the task-level: source segment + target segment + projectToken + userEvaluating + uniqueQuestionControlHash This hash may be useful to determine whether a specific segment has already been evaluated for a specific question by a given evaluator. |
| Var7 | Username of provider of evaluation score (i.e., evaluator) | |
| Var8 | Source segment index | |
| Var9 | Duplication logic value | 0= Not set 1=Avoid duplication at task level 2=Avoid duplication at project level |

Table 2: Evaluation metadata

6 Deployment Guidelines

The ACCEPT framework is extremely flexible, which means that entities can deploy some of its components in multiple ways.

6.1 Possible Use Cases

For instance, an entity that is interested in making some pre-editing functionality available to its users outside of the Portal would need to deploy:

- The ACCEPT API and framework
- The Pre-Edit section of the ACCEPT necessary to provide pre-editing functionality outside of the Portal (e.g., application key management)
- The Pre-Edit client.

The back-end components (the ACCEPT API, framework and Portal) would then have to be deployed first, before the front-end component (i.e., the Pre-Edit client) can be deployed in a given user environment (e.g., a forum platform, a CMS, etc.). The front-end component would have to be configured to point to the ACCEPT API deployed in the first step.

If an entity were interested in making some post-editing functionality available to its users outside of the Portal while evaluating the post-editing quality generated using a different set of users, the following components could be deployed:

- The ACCEPT API and framework
- The Post-Edit and Evaluation sections of the ACCEPT Portal⁷
- The Post-Edit client.

The back-end components (the ACCEPT API, framework and Portal) would then have to be deployed first, before the front-end component (i.e., the Post-Edit client) can be deployed in a given user environment (e.g., a forum platform, a CMS, etc.). The front-end component would have to be configured to point to the ACCEPT API deployed in the first step.

With so many possible deployment scenarios, some general deployment guidelines are provided in the next section. These guidelines are not expected to replace the detailed deployment instructions provided in the various Github.com repositories (e.g., [PreEdit clients](#), [Post-Edit client](#), [Evaluation client](#), [Portal](#), [API](#)).

6.2 High-level Deployment Guidelines

In general, the following deployment guidelines apply:

- The ACCEPT API depends on the ACCEPT Framework since it is the Framework that is responsible for data persistence.
- The ACCEPT Portal depends on the ACCEPT API since the Portal uses the API for all operations, from user authentication to projects creation.
- The ACCEPT JavaScript clients depend on the ACCEPT API since information flows constantly from both sides.

⁷ These Portal sections are necessary because the API itself does not provide the complete required functionality. For example, post-editing and evaluation projects can currently only be created via the Portal.

Given the dependency scenario above, the ACCEPT system should be deployed as follows:

1. The ACCEPT API and the ACCEPT Framework;
2. The ACCEPT Portal;
3. The jQuery plug-in(s) implementation.

As mentioned above, the step by step deployment for each component can be found within each repository's core readme file. These steps include information on how to meet the .NET/C# (mostly using the [NuGet package manager](#)) and JavaScript dependencies. For the deployment of the first three components (ACCEPT API, ACCEPT Framework and ACCEPT Portal), the following server configuration was used during the project execution period:⁸

- Windows Server 2008 R2 x64 as Operational System
- 16 GB of RAM
- 30 GB of Disc Space
- Processor Xeon E52570 @2.5 Ghz (4 Cores)
- Microsoft .Net Framework 4.5 installed
- IIS (Internet Information Services) 7.5
- SQL Server 2008 R2 or other (MySQL for instance, but additional configuration steps might be needed) as DB technology.

After deployment, the ACCEPT components still need some extra manual configuration input, as described in the following sections.

6.2.1 Server-Side

For the server-side components, configuration options can be specified within specific configuration files (e.g., [ACCEPT API](#), [ACCEPT Portal](#)).⁹ The configuration entries can be found in the "appSettings" element, as shown in the following examples.

ACCEPT API Example 1

```
<!--Trigger Database schema operations. Accepts the values: "0" for disabled and "1"
for enabled-->
<!--Drops the current Database schema.-->
<add key="DbShemaDrop" value="0"/>
<!--Creates the current Database schema.-->
<add key="DbShemaCreate" value="0"/>
<!--Updates the current Database schema.-->
<add key="DbShemaUpdate" value="0"/>
```

This example shows three configuration entries that can be used to trigger actions on the database's side. In this case, manually tweaking the value between one and zero would respectively trigger or stop the commented action.

⁸ This configuration is not considered minimal or ideal – it states the configuration used during the studies that took place within the scope of the ACCEPT project.

⁹ Informational comments can be found at the top of each application's configuration entry.

ACCEPT API Example 2

The ACCEPT system uses an Object Relational Mapping ([ORM](#)) framework to persist data, currently [Fluent NHibernate](#) (version 3.2). This means that the ACCEPT System is independent from the database technology placed underneath the ORM. The database technology can be tweaked within the ACCEPT API's configuration file:

```
<hibernate-configuration xmlns="urn:nhibernate-configuration-2.2">

    <session-factory>

        <property
            name="connection.driver_class">NHibernate.Driver.SqlClientDriver</property>
        <property name="connection.connection_string">Data
            Source=DB_SERVER_NETWORK_PATH;Initial Catalog=DB_NAME;Integrated
            Security=True</property>

        <property name="adonet.batch_size">10</property>

        <property name="show_sql">>true</property>

        <property name="dialect">NHibernate.Dialect.MsSql2008Dialect</property>
        <property name="use_outer_join">>true</property>

        <property name="command_timeout">60</property>

        <property
            name="proxyfactory.factory_class">NHibernate.ByteCode.LinFu.ProxyFactoryFactory,
            NHibernate.ByteCode.LinFu</property>

    </session-factory>

</hibernate-configuration>
```

Changing the value of the highlighted property in the example above allows for a re-configuration of NHibernate to communicate with specific database technologies.¹⁰ Examples include:

SQL Server 2012: `<property name="dialect">NHibernate.Dialect.MsSql2012Dialect</property>`

MySQL: `<property name="dialect">NHibernate.Dialect.MySQLDialect</property>`

DB2: `<property name="dialect">NHibernate.Dialect.DB2Dialect</property>`

The *connection* string property may vary depending on the underlying database technology, so different database technologies may have different connection strings too.¹¹

¹⁰ More information on NHibernate configuration process can be found at <http://nhforge.org> (accessed: December 2014).

¹¹ More information on the structure of connection strings per technology can be found at <http://www.connectionstrings.com> (accessed: December 2014).

ACCEPT Portal Example

```
<!--The URL for the ACCEPT API endpoint-->
```

```
<add key="AcceptPortalApiPath" value="ADD_ACCEPT_API_ENDPOINT_URL_HERE" />
```

This example shows one of the most important configuration entries for the ACCEPT (Web) Portal: the ACCEPT API URL. As stated before, the ACCEPT Portal depends on the ACCEPT API for performing all data persistence operations, which means this entry is mandatory for establishing the communication between both components.

6.2.2 Client-Side

Once clients have been downloaded and dependencies have been met, the ACCEPT jQuery clients also need some extra manual configuration input. As described in previous deliverables (e.g., [Deliverable D5.6](#)) the configuration entries can be provided at instantiation stage (i.e., when the plug-in is actually initialized). Further explanations and some context on each configuration entry can be found within each plug-in's main readme file.

7 Conclusion

This deliverable described the work that was required to turn the post-editing and evaluation prototypes developed in Years 1 and 2 into full-fledged demonstrators. Specifically, the post-editing client prototype was enhanced by integrating the resources and technologies developed in WPs 2 and 7 (i.e., post-editing rules and paraphrasing service). The evaluation environment prototype was also enhanced in order to allow existing post-edited content to be seamlessly evaluated using the ACCEPT Portal. All software code described in this deliverable has been made publicly available under an Apache v2 license on the Github.com platform (alongside detailed deployment instructions that supplement the high-level guidelines included in the present deliverable).

References

- Mitchell, Linda, Johann Roturier, David Silva:
Using the ACCEPT framework to conduct an online community-based translation evaluation study.
In: *Proceedings of the Seventeenth Annual Conference of the European Association for Machine Translation (EAMT)*, pages 155–158, Dubrovnik, Croatia, June 2014.