



Published on Multilingual Online Translation (<http://www.molto-project.eu>)

D3.3 Translation Tools – Workflow Manual

Contract No.:	FP7-ICT-247914
Project full title:	MOLTO - Multilingual Online Translation
Deliverable:	D3.3 MOLTO translation tools – workflow manual
Security (distribution level):	Public
Contractual date of delivery:	M31
Actual date of delivery:	April 2013
Type:	Manual
Status & version:	Final
Author(s):	Inari Listenmaa, Jussi Rautio
Task responsible:	UHEL
Other contributors:	Lauri Alanko, John Camilleri, Thomas Hallgren

Abstract

Deliverable D3.3 consists of a manual and a description of the workflow of the MOLTO translation tools. The document introduces the components: the open-source translation management system [Pootle](#), the [Simple Translation Tool](#), which supports many different translation methods and the [Syntax Editor](#), which allows to modify text by manipulating abstract syntax trees.

The document presents two translation workflows. The first scenario integrates MOLTO tools in a professional translation on fixed source, using Pootle. MOLTO translations with GF grammars are added in machine translation options. Another direction taken is the population of translation memory with GF generated data. In the second workflow, the translator is authorized to do changes to source. The tools used in this scenario are the Simple Translation Tool and the Syntax Editor.

1. Introduction

This Deliverable D3.3, is a manual and a description of workflow for the translation tools produced within [WP3](#). As stated in the previous Deliverables [3.2](#) and [3.1](#), the user of the translation tools is not required to know how to write GF grammars. They are either translators, whose job is to translate from fixed source, or they are authorized to modify the source text in order to fit into the structures covered by the domain-specific grammar(s). This document presents workflows for both scenarios.

1.1 Outline of the document

In Section 2, we present the components: the open-source translation management system [Pootle](#), the [Simple Translation Tool](#) and the [Syntax Editor](#). In Section 3, we present the translation workflows. In Section 4 we talk about future work, such as translation memory generation and further integration of the GF cloud tools.

1.2 Changes from Deliverables 3.1 and 3.2

We have changed the plan from Deliverables D3.1 (translation tools API) and [D3.2](#) (prototype). The previous deliverables use GlobalSight, a translation management system, and an external editor that supports GF. However, we found that GlobalSight was not maintained, whereas Pootle is a modern and lightweight translation platform, with an active user base. Following the recommendation of a professional translator, we decided to integrate the GF translation to Pootle instead.

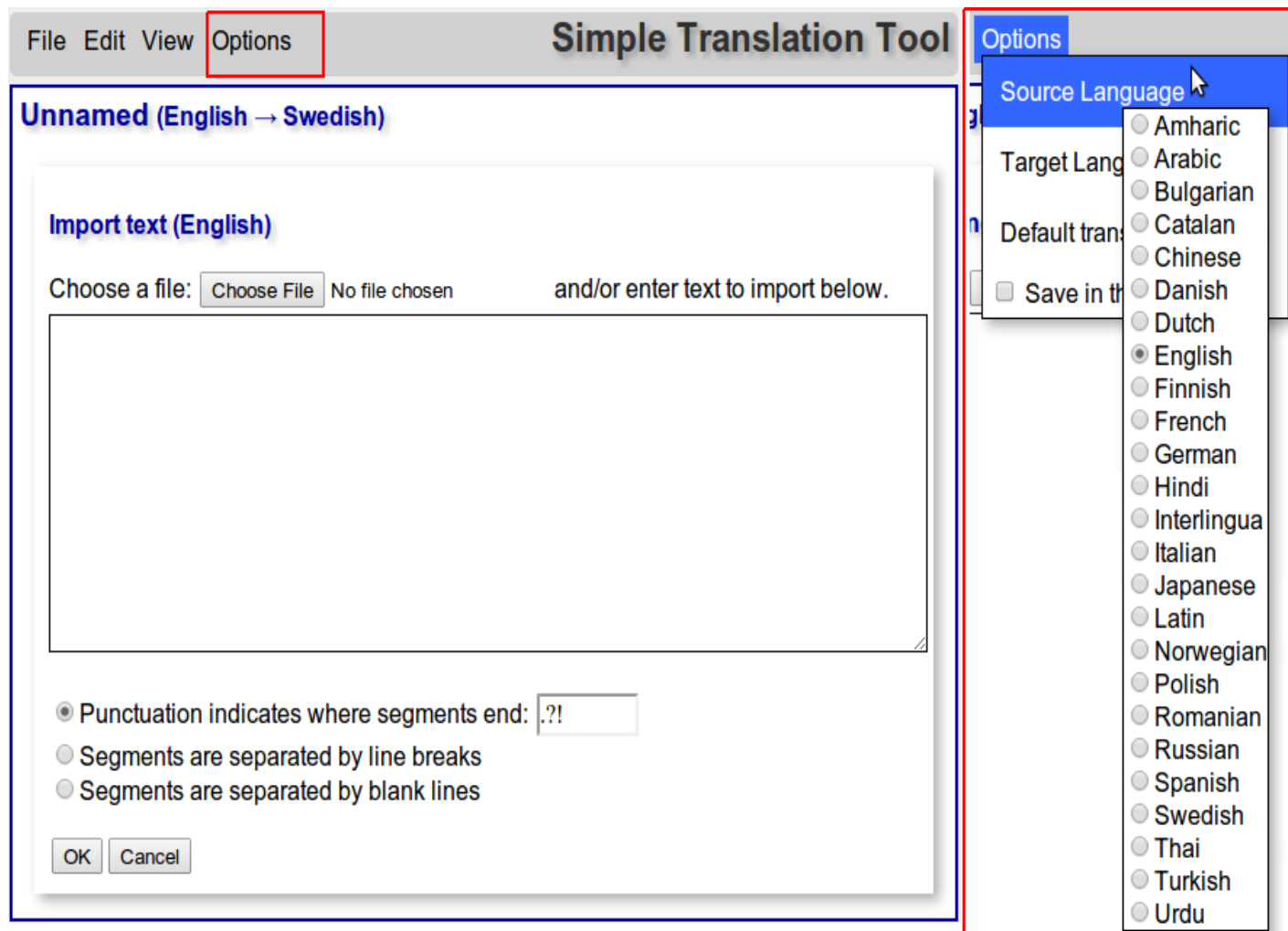
Due to the changes in the plan, we do not include a summary of the Deliverables 3.1 and 3.2. The sections of this document are self-contained, assuming that the reader is in general familiar with MOLTO project.

2. Components

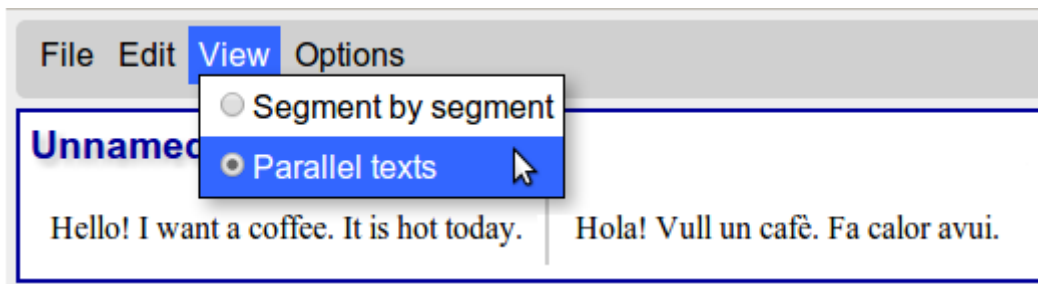
2.1 Simple Translation Tool

The Simple Translation Tool (<http://cloud.grammaticalframework.org/translator>) is a translator's editor that supports manual and automatic translation. Documents consist of a sequence of segments that are translated independently. The user can import text in the source language and obtain automatically translated text in the target language. Imported text can be segmented based on punctuation. Optionally, one can also use line breaks or blank lines to indicate segmentation in imported text. Text can be edited after it has been imported.

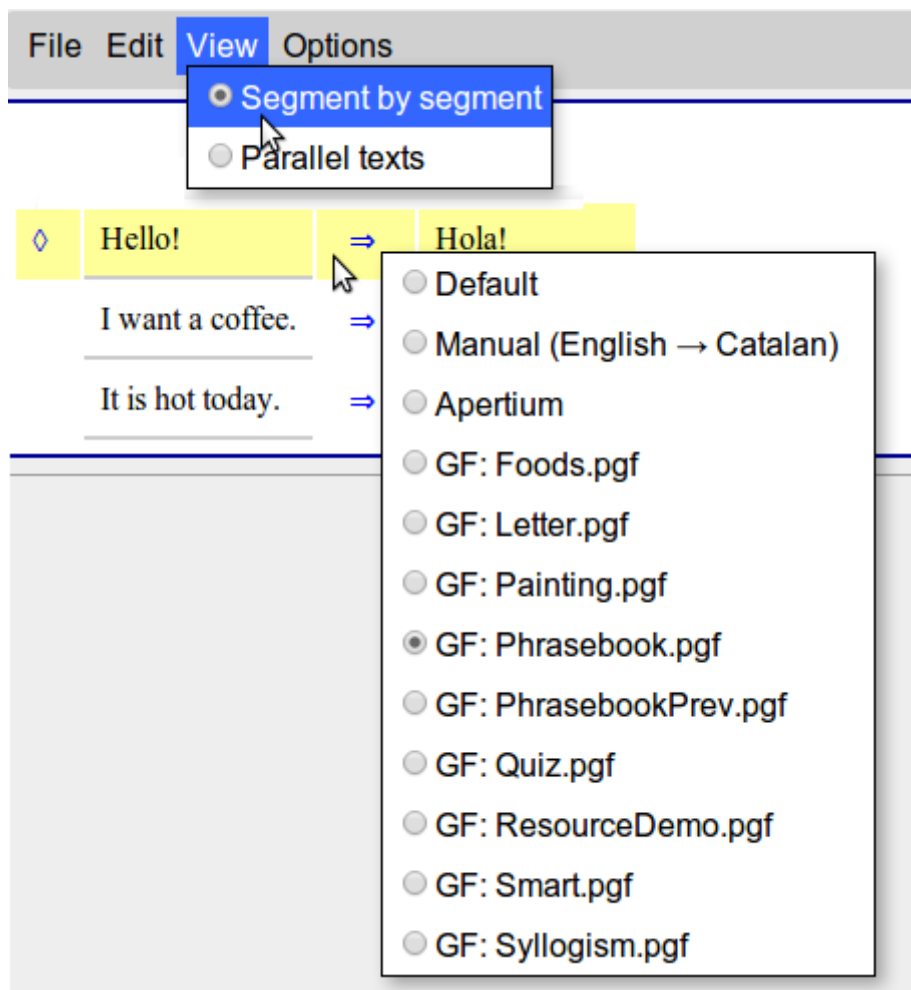
In the image below, the translator chooses the source and target languages and uploads a text in the source language.



The text can be displayed as parallel texts or segment by segment, as shown on the next page.



Option 1: Parallel texts



Option 2: Segment by segment

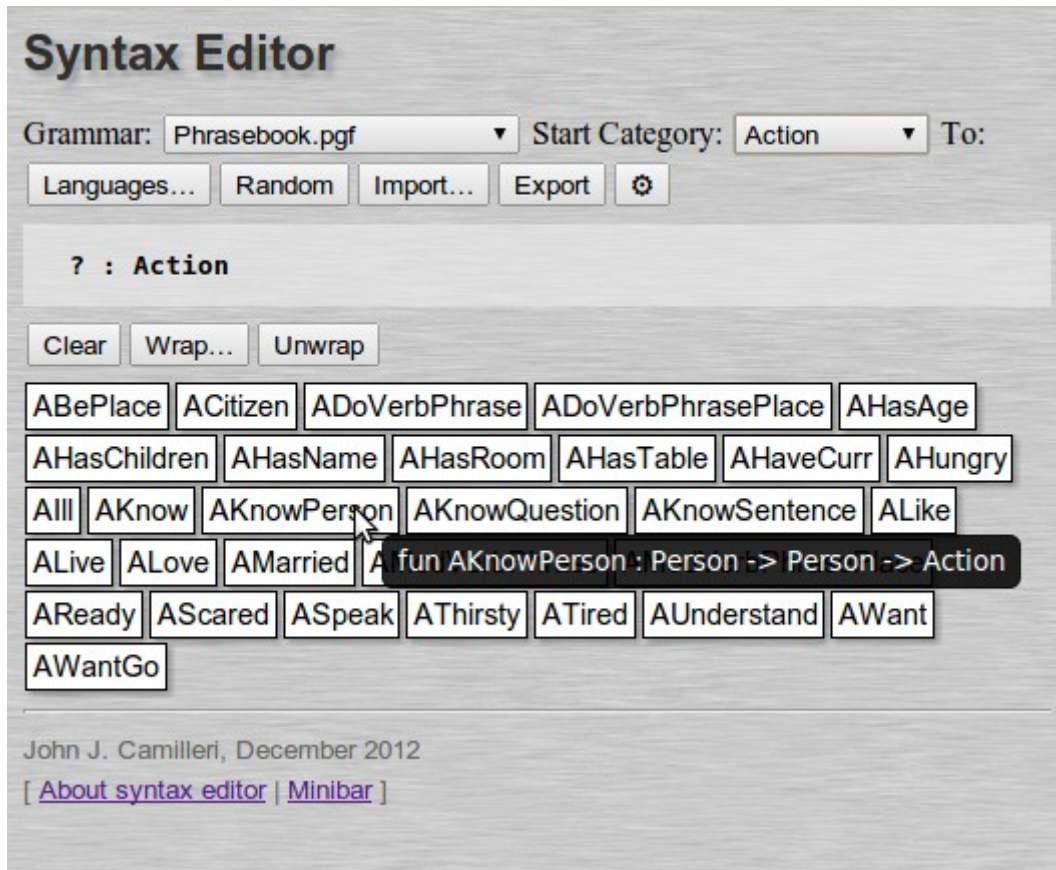
The translator can choose a translation method for the whole document, or when needed, for each segment. The translation methods include various GF grammars, transfer-based machine translation by Apertium and manual translation. Other machine translation options can be added as well.

The choice of grammars is shown in the picture above. Choosing a different grammar for different segments is relevant, if the body of the text is unrestricted, but there are passages where precision is required, such as formulas in a patent application. Then the unrestricted text can be translated with a method that has more coverage but less precision, and the formulaic parts with a specialized grammar.

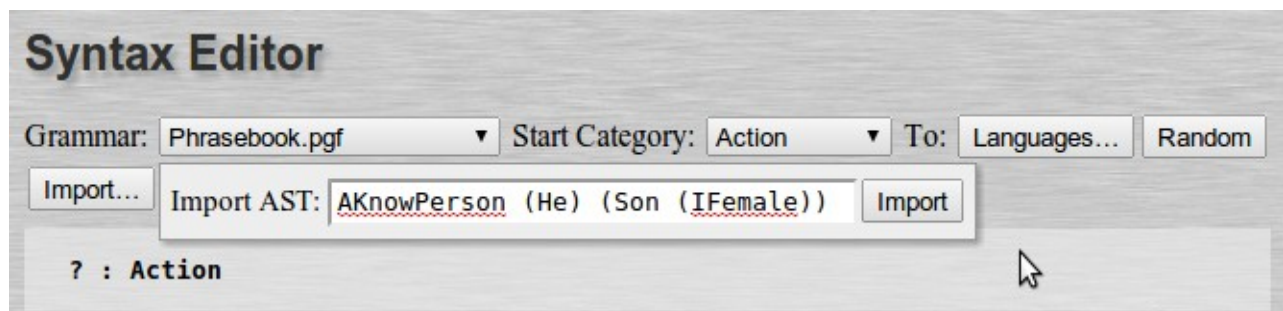
2.2 Syntax Editor

The Syntax Editor (<http://cloud.grammaticalframework.org/syntax-editor/editor.html>), written by John Camilleri, is a tool for building and manipulating abstract syntax trees in GF.

The image below shows an initial view of the Syntax Editor. The chosen grammar is *Phrasebook* and the start category is *Action*; a verb phrase whose tense and polarity are not fixed yet. This excludes some of the possibilities to construct a sentence in *Phrasebook*, for instance greetings and other fixed phrases. Choosing another start category would give these options as well, *Action* is chosen here for the sake of simplicity.

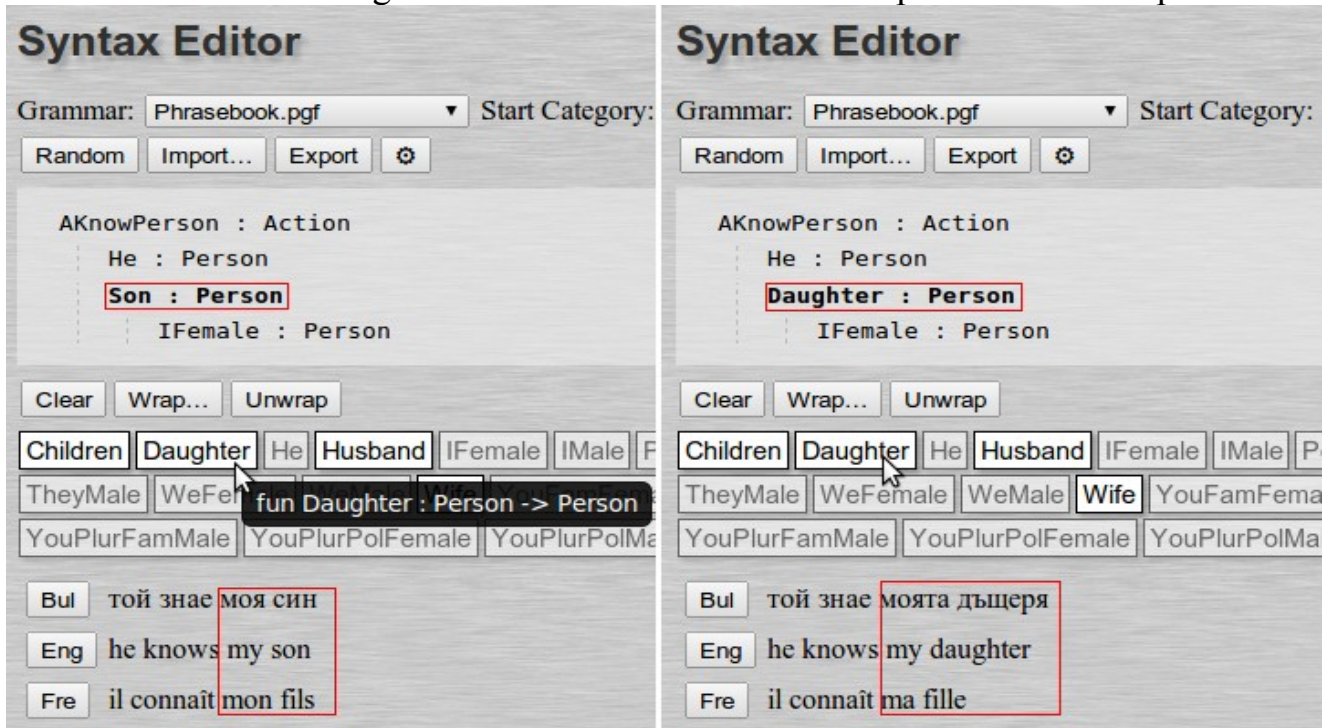


A syntax tree can also be imported as raw text, as shown in the image below.

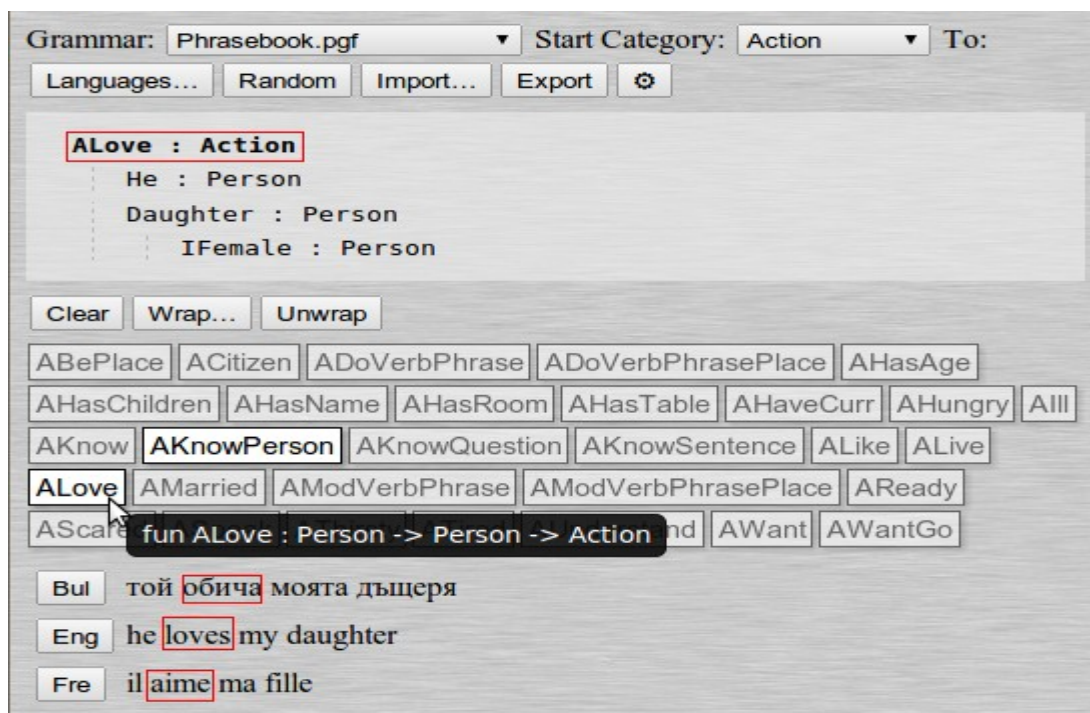


The editor maintains the structure of the abstract syntax tree (AST) and outputs linearizations in the languages of the concrete syntaxes. In the following three

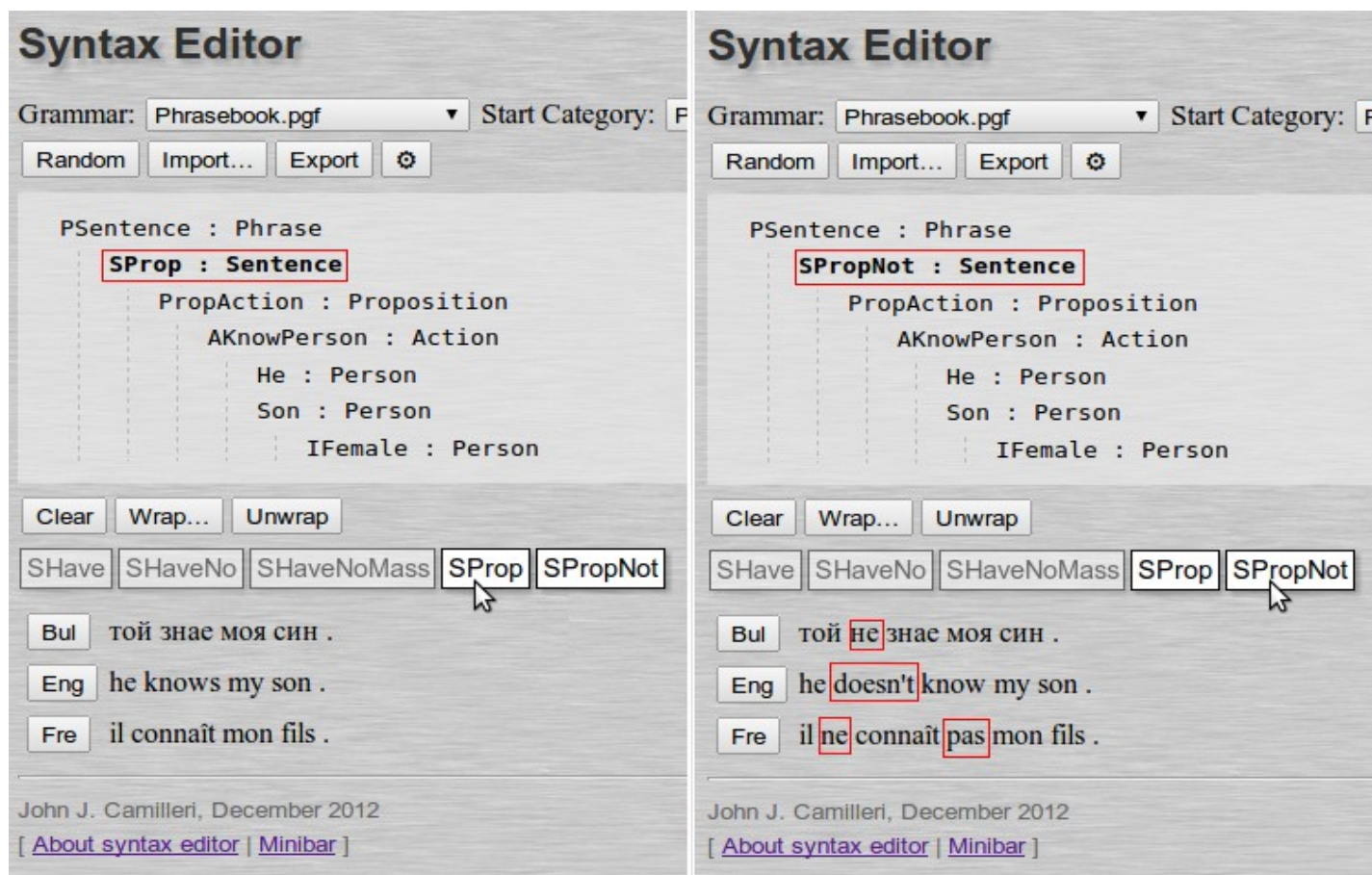
images, the first shows a change of an argument in the AST and the consequent change in the linearizations. Son is changed to a daughter, in English only the word changes, but in French and Bulgarian, changing the gender of the argument also affects the agreement of the possessive pronoun.



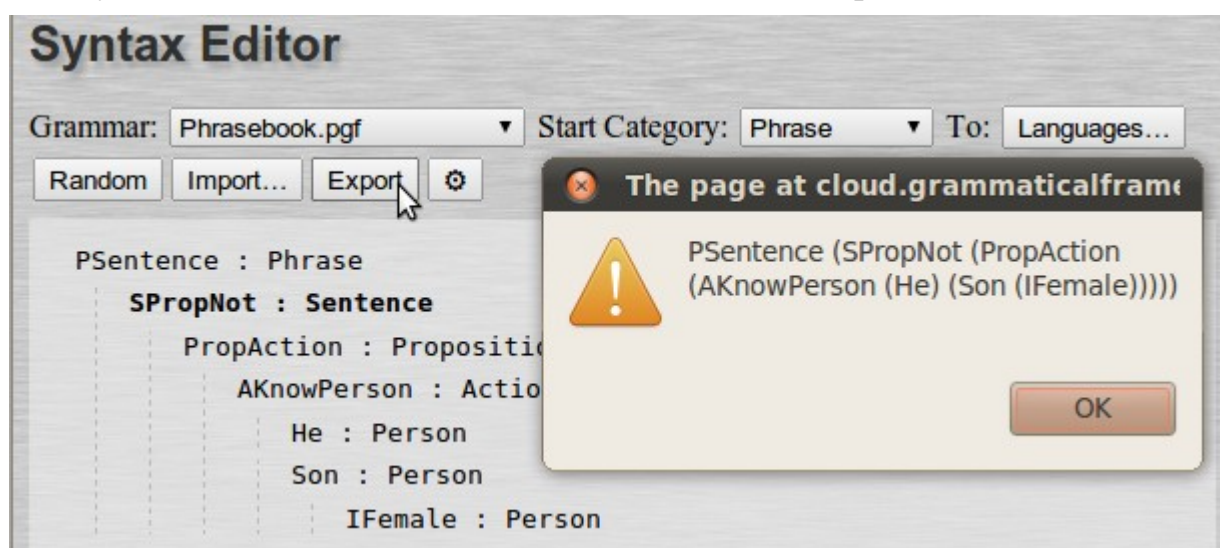
The second image shows the change of the head of the phrase. Both `AKnowPerson` and `ALove` take two arguments of type `Person`, so from the point of view of the AST, they are interchangeable.



The third image manipulates the polarity of the sentence. As opposed to the previous two examples, this AST is complete, belonging to the start category *Phrase*. This start category includes information whether the action is affirmative or negative, and uses constructors such as *SProp* or *SPropNot*.



Finally, a tree created or modified in the editor can be exported as raw text.



There is a work in progress to make Syntax Editor accessible from the Simple Translation Tool, as a way to modify the source text. More details on the process are explained in Sections 3.2 and 4.2.

2.3 Pootle as an example of GF integration

Pootle (<http://pootle.translatehouse.org/>) is an open-source translation and project management platform that is completely implemented as a Web service. The platform accepts most of the formats used in translation industry like XLIFF, TMX and PO.

2.3.1 Integration of GF into Pootle

Pootle has a support for both standard translation memories and machine translation. Google Translate and Apertium MT systems are available in the standard installation, and Web server queries to other MT systems can be sent by modifying the source code of Pootle. GF translation via GF Web API has been added to the Pootle translation environment as a proof of concept.

Unlike Google Translate and Apertium which only require the translatable segment and the source and target language codes as input, the GF system also needs the name of the resource grammar(s) to be used. The selection of the grammar has been added to the Pootle project administration dialog, where the project manager normally selects the languages, file formats, translation memories and other resources to be used:

Pootle Demo jussi Projects

Administration / Projects

Info General Users Languages Projects Permissions

Here you can add, edit, and delete all the available projects in this Pootle server.

Code	Full Name	Quality Checks	File Type	Source Language	GF Grammars	Ignore
FR_XLIFF	XLIFF in French	standard	XLIFF	French - fr	Phrasebook Resource Demo	<input type="checkbox"/>
combined	Phrasebook+Resource	standard	Java Property	English - en	Phrasebook Resource Demo	<input type="checkbox"/>
terminology	Terminology	terminology	Gettext PO	English - en	Phrasebook Resource Demo	<input type="checkbox"/>
SV_XLIFF	XLIFF in Swedish	standard	XLIFF	Swedish - sv	Phrasebook Resource Demo	<input type="checkbox"/>

Code Full Name Quality Checks File Type Source Language GF Grammars Ignore

Save Changes

Pootle admin view

The translator user interface shows the buttons for different MT systems above the edit box. On the screenshot below, there are buttons for three MT engines: GF, Google and Apertium.

The screenshot displays the Pootle Demo web interface. At the top, a blue header bar contains the Pootle logo, the text "Pootle Demo", and user navigation links: "jussi", "Projects", "ADMIN", and "Log Out". Below the header, a breadcrumb trail reads "Projects / Phrasebook+ResourceDemo / Swedish". A secondary navigation bar includes tabs for "Overview", "News +", "Translate" (which is active), "Terminology", and "Permissions". The main content area shows a list of translation segments. The first segment is "This pizza is cold .". The second segment is also "This pizza is cold .". The third segment is "The house is red .". The fourth segment is "The house is red .". The fifth segment is "Is the milk warm ?". The sixth segment is "Where is the museum ?". This segment is expanded, showing a text input box with the Swedish translation "var är museet ?". Above the input box, there are icons for "English", "G" (Google), "A" (Apertium), and a "GF" (Grammatical Framework) icon. A button labeled "Grammatical Framework (EN→SV)" is positioned to the right of the input box. Below the input box is an "Add Comment" link. To the right of the input box is a green "Submit" button and a checkbox labeled "Needs work".

When the GF button is clicked, the web browser sends the source segment to the GF server, which parses and linearizes it into the target language using the given grammar(s). The translated text is then sent back to the browser. The translation suggestion can then be edited, after which the segment is added to the project translation memory.

This illustrates that the GF translation system can be relatively easily added into various translation editing applications via the Web API. The GF Web API can be modified to comply with any standard Web API like the one proposed by TAUS (<https://tauslabs.com/interoperability/taus-translation-api>).

2.3.2 Installation instructions

The modified Pootle source code is available as a Git repository at <http://tfs.cc/git/pootle.git>. The repository contains general Pootle documentation and the GF specific instructions. To enable the GF support, add the tuple ('GF', '') to the `MT_BACKENDS` configuration variable in `pootle/settings/60-translation.conf`.

For the moment, the list of available GF grammars is hard-coded in Pootle. Modify the function `pootle.apps.pootle_app.management.create_sample_grammars` to change the list of grammars created for a new database. After database creation, new grammars have to be added manually using the `pootle_project.models.Grammar` model interface. Each grammar must be associated with a URL that serves translation requests using the `pgf-http` API.

Once Pootle has a list of available grammars, they can be allocated to projects in Pootle's web administration interface. In the `ADMIN->Projects` tab, simply edit the `GF Grammars` column for each project.

3. Workflows

This part consists of two separate workflow descriptions. The first workflow is that of a traditional professional translation, using the translation platform Pootle, with GF integrated in machine translation and translation memory. The second workflow describes a case where the translator is authorized to do changes to source. The tools of choice are the Simple Translation Tool and the Syntax Editor.

3.1. Translation of a fixed source

The workflow of a professional translation is often fairly complex, including roles such as project manager, translator and reviewers of both content and language. Machine translation is used as the translator's aid, along with other tools such as dictionaries and translation memories. This is an established practice in computer-assisted translation; one of the main objectives of [WP3](#) is to demonstrate that MOLTO tools can be adapted in a traditional translation workflow.

The translator is not allowed to modify the source text, which is a serious limitation for the MOLTO translation, precision at the cost of coverage. However, in this scenario we assume that the translator necessarily knows both source and target languages. The role of the machine translation is not to provide publication quality text for blind translation, but to help the translator to produce translations.

When does it make sense to use MOLTO tools? With free text of unrestricted domains, the most common case is that GF grammars do not produce any translation at all, due to missing words or constructions. When we have a professional translator post-editing GF grammars beat general-purpose MT in situations where the structure is crucial. Any formulaic parts within unrestricted text, such as mathematical constructions (case study, WP6) and chemical formulas (case study, WP7). A formula such as *(2S)-2-[(4S)-4-(2,2-difluorovinyl)-2-oxopyrrolidinyl]butanamide* is constructed by elaborate rules, which can be expressed precisely with a GF grammar. However, statistical machine translation fails to capture the structure, and the result is worthless for post-editing; a change, addition or omission of even one element is enough to change the formula completely.

Thus, we have integrated GF as one of the translation options in [Pootle](#). The technical side is handled by GF Web API calls, explained in Section 2.3.

3.1.1 The Pootle workflow

The Pootle translation environment implements the now industry-standard workflow where the translatable material, the translation memories and the editing tools reside on the same Web server. The system also includes real-time word-count reporting, user management and terminology asset handling. This greatly reduces the effort needed for a translation project, as all the tools and resources are centralized. As the translations are updated into a shared translation memory in real time, the need to create, update and document the memories after the project is unnecessary. Pootle

also allows the local downloading of necessary resources in cases where the translator does not have an always-on Internet connection.

3.1.2 Translation project manager

The translation project manager can upload the files to be translated to the Pootle server and define the language pairs, translation memories, glossaries (either general or project specific) and file formats to be used in the translation. The systems allows the use of standard translation file formats like XLIFF (for source material), TMX (translation memories) and TBX (terminology). In our GF machine translation enabled version of Pootle, the PM can also select the GF grammar or grammars used for translation (see Section 2.3 for an example).

When the translation assets have been configured, the PM can give the necessary access rights to them for the translators and reviewers. The material can also be translated as crowd-sourcing, so anyone with an access to the Pootle server can participate in the translation. This method has been used in many open-source localization projects, for example in the OpenOffice suite.

3.1.3 Translators

The translators can then log in with their credentials onto the Pootle server, and see all the translation tasks assigned to them by the PM:

Projects

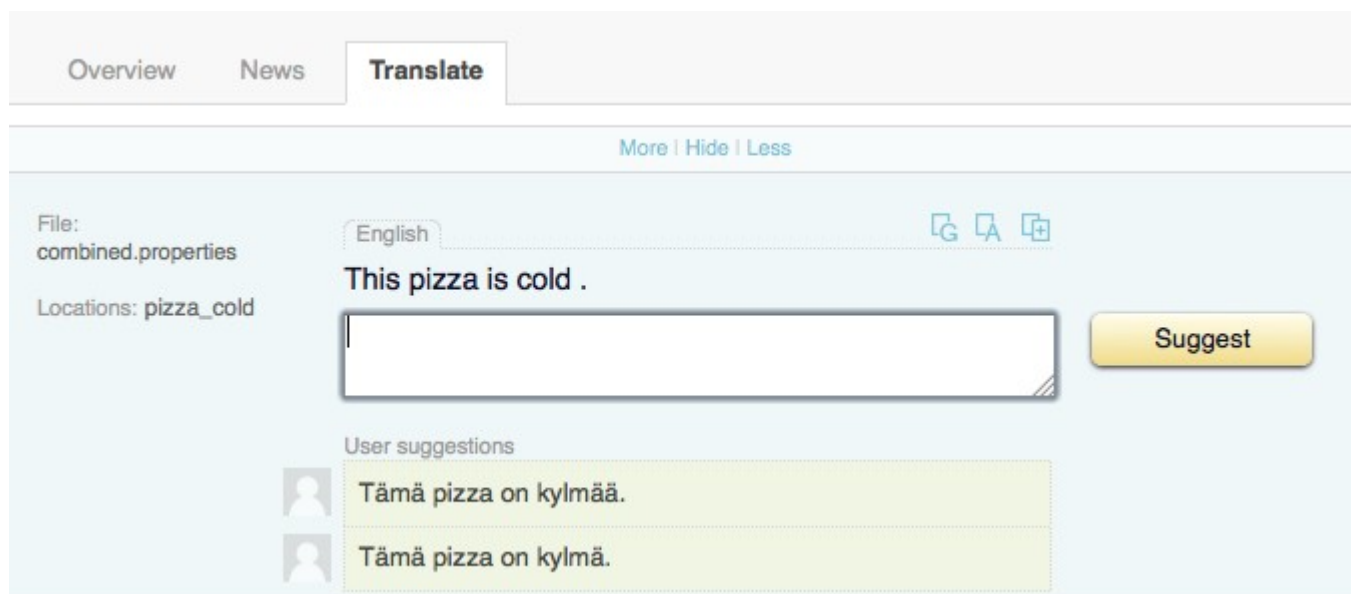
Project ↕	Progress ↕	Last Activity ↕
Accentuate.us	<div><div></div></div>	2010-10-15 19:17 (scannell)
ANLoc Formation FR	<div><div></div></div>	2013-03-26 22:09 (oldking)
Art of Illusion	<div><div></div></div>	2013-04-11 07:46 (mponiek)
BirdFont	<div><div></div></div>	2013-03-28 02:29 (nobody)
FileZilla	<div><div></div></div>	2013-03-23 22:31 (oldking)

After clicking a project name, the `Languages` page shows the target languages the translator has been assigned to:

21 languages, 69% translated

Name ↕	Progress ↕	Total ↕	Need Translation ↕	Last Activity ↕
Afrikaans	<div><div></div></div>	8930	7164	2010-08-05 10:03 (admin)
Arabic	<div><div></div></div>	8933	0	2010-10-31 18:02 (haytham9687)
Catalan	<div><div></div></div>	9207	0	2010-06-05 17:47 (ellibre)
Catalan (Valencia)	<div><div></div></div>	9207	1066	2010-05-30 04:38 (PauGNU)

Clicking the language name opens the editing environment:



Any exact or fuzzy match (“Translation suggestion” in Pootle terminology) found in the translation memory can be selected and edited for a translation. As explained in the previous section, the translator can use machine translation services (including GF) by clicking the relevant button. The translator is thus able to use either the suggestions from the translation memory, a choice of machine translations or a translate the segment from scratch.

The Pootle editing tool includes automatic checking for quality issues, for example missing tags, variables or numbers, wrong capitalization, punctuation and so on, so the translator gets instant feedback on possible formatting errors in the translation. The translator is also able to include comments to reviewers and PMs as separate field in the tool and add or review terms in the terminology.

3.1.4 Review and post-processing

During the translation, the PM can follow the progress of the project on the Projects page. When the translation of a component is ready, the PM gets a notification, and the translation can be sent to reviewers, who then check and correct the translations by accepting or rejecting the suggestions. After the review process, the PM is again notified, and the translated and reviewed file can be downloaded for post-processing.

3.2. Translation of an editable source

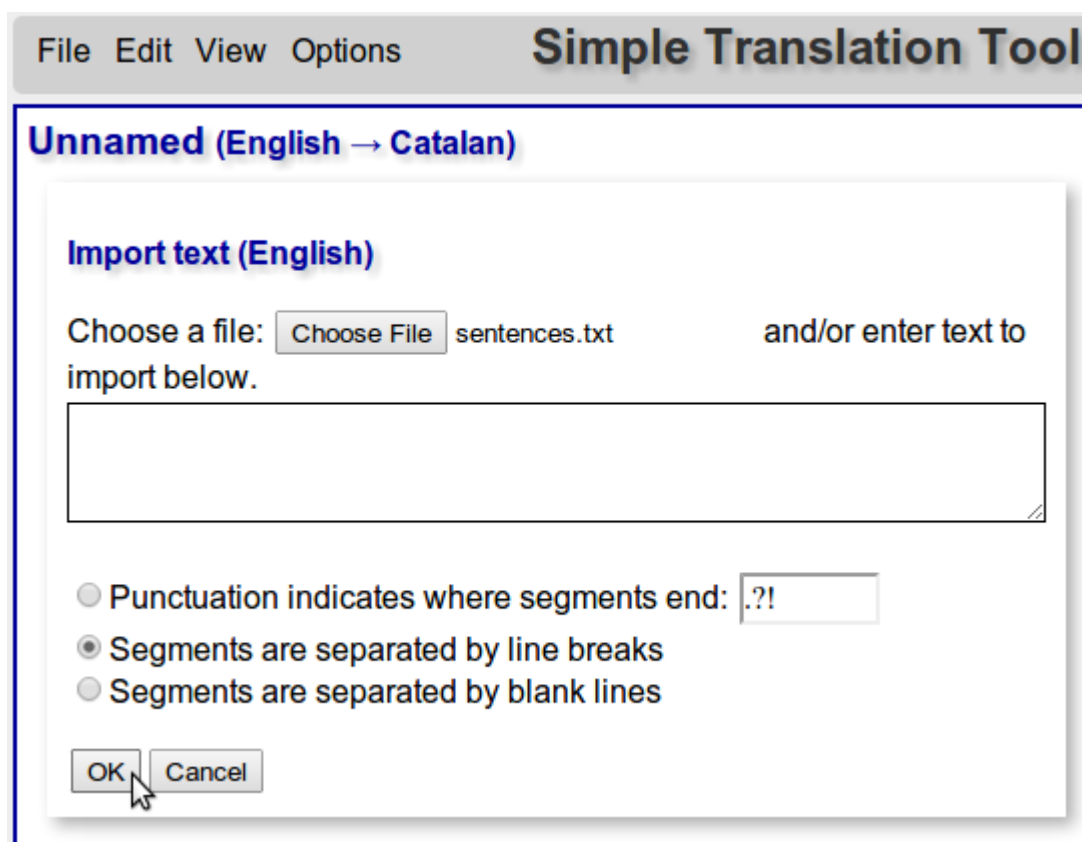
We hold on to the assumptions stated in [D3.1](#):

- Translator is the author or is authorized to adapt the text to better satisfy the constraints of the translation grammar,
- Translator is native or fluent in the source language, and familiar with the domain or at least its special language in order to know how the message can be paraphrased,
- Translator is not required to know (all of) the target languages.

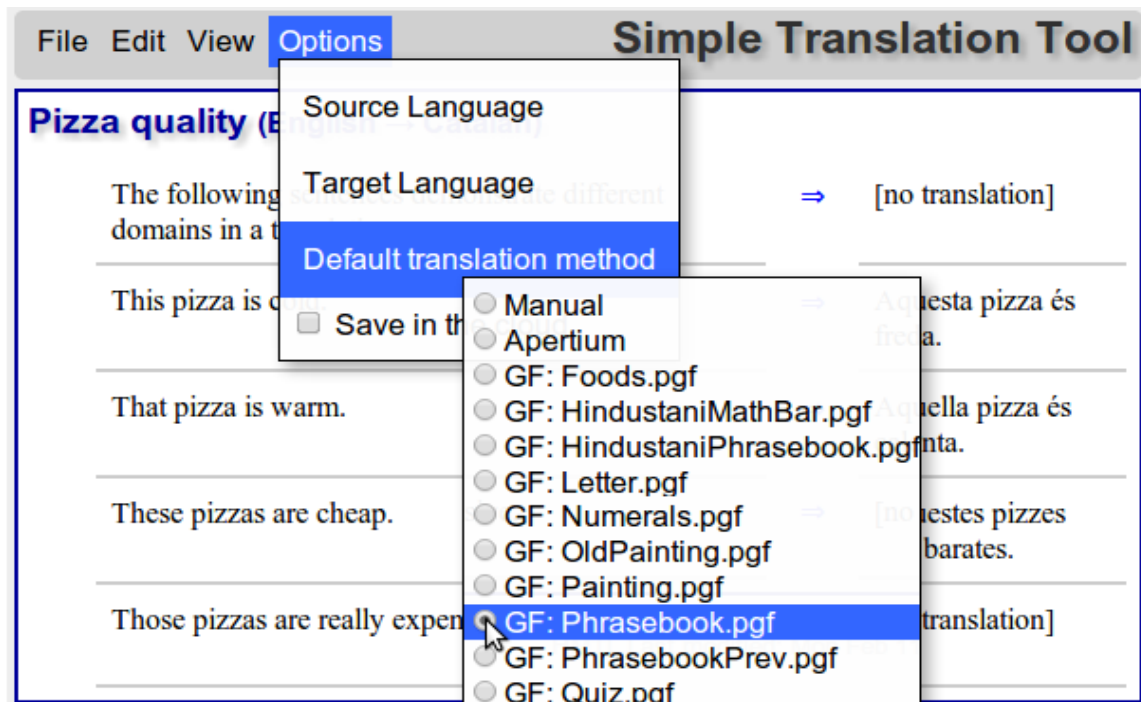
In a case of at least partially blind translation, the quality of MT needs to be excellent. External revisers can be added to this scenario as well, but we assume the quality to be in general good, errors are due to bugs in grammars and grammar writers are correcting them. A concrete scenario could be a multilingual website, where the authorized users can create content in any of the languages, and it is updated simultaneously into all of them. Assuming there are users for every language, they can work themselves as reviewers, providing feedback in case there is an error in the grammar. Then a grammar writer fixes the grammar, and the all structures that had the same problem will be updated.

There can be an existing source document or it can be created from scratch. In any case, there is a need for guided authoring, to ensure that the produced text is recognized by the grammar. This is not currently implemented, but planned by UGOT and explained further in Section 4.2.

The Simple Translation Tool (STT) offers the functionalities for pre- and post-editing of MT. When needed, machine translation can be completely overridden by manual translation. The functionalities of STT are demonstrated with a toy text about pizzas. More complex grammars produced within MOLTO include the patent grammars in [WP7](#) and the mathematical grammar library in [WP6](#), but they are not integrated to STT at the moment. We plan to produce a video demo with some real use cases. In the screenshots below, the text is uploaded into STT and a default translation method is chosen.

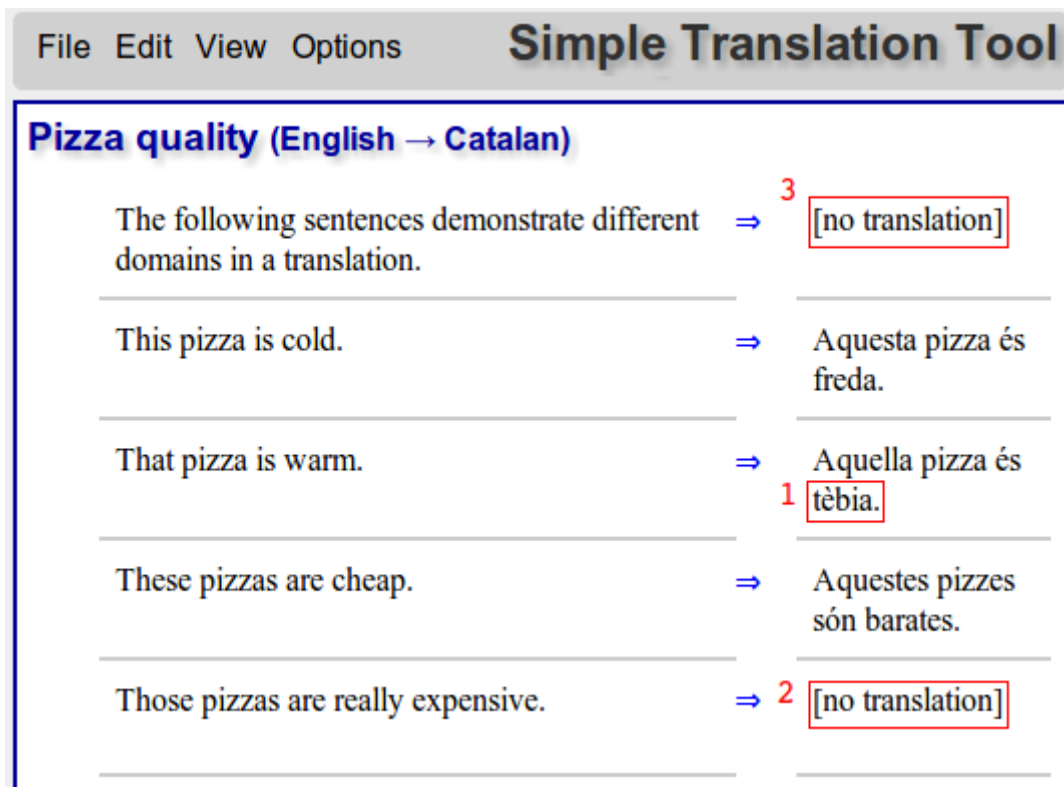


1. Uploading text in STT

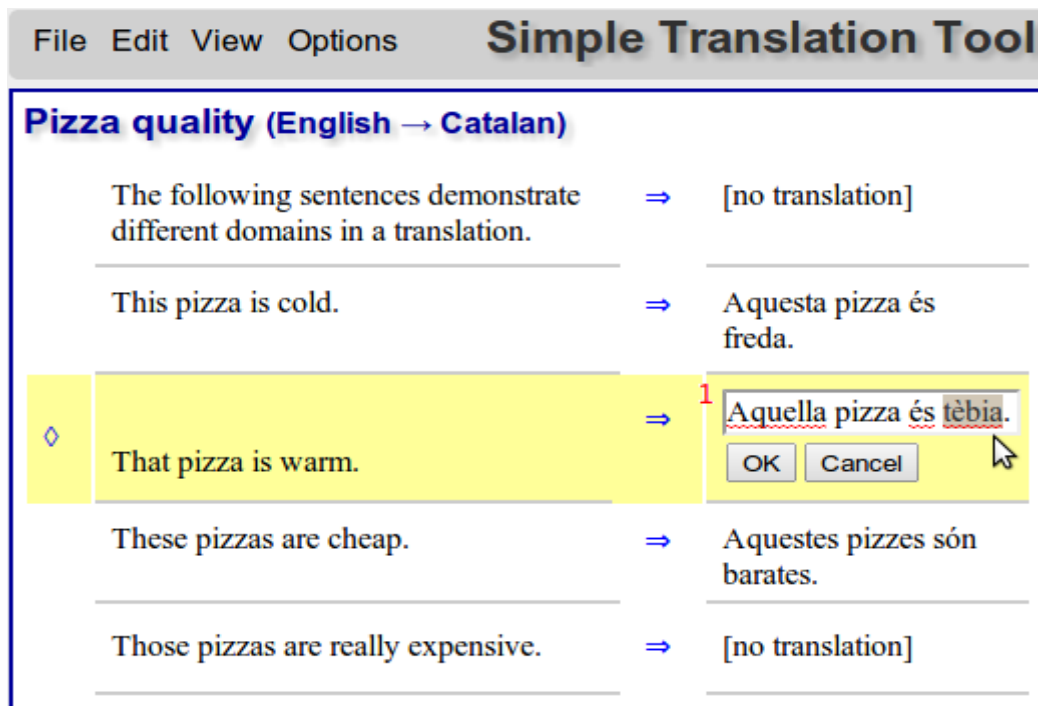


2. Choosing default translation method

In the third image, we see three errors. The first one, indicated with number 1, is an error in the grammar; an unidiomatic word choice. This type of error is easiest to fix just by modifying the target, as shown in the fourth image – followed by a bug report to a grammar writer. Of course, spotting this type of error requires that the translator knows the target language. In a case where they do not know the target language, they just need to fix the cases where a sentence fails to translate at all.

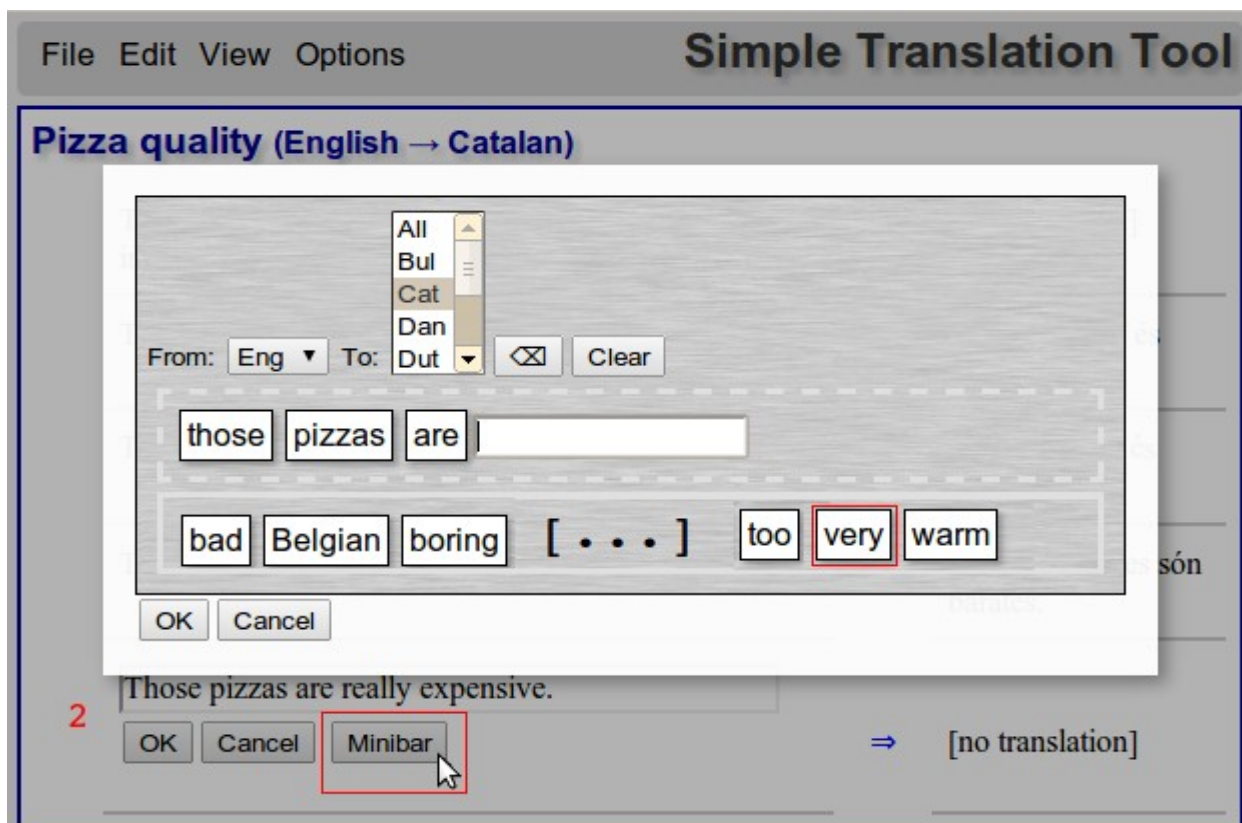


3. Three errors in the translation



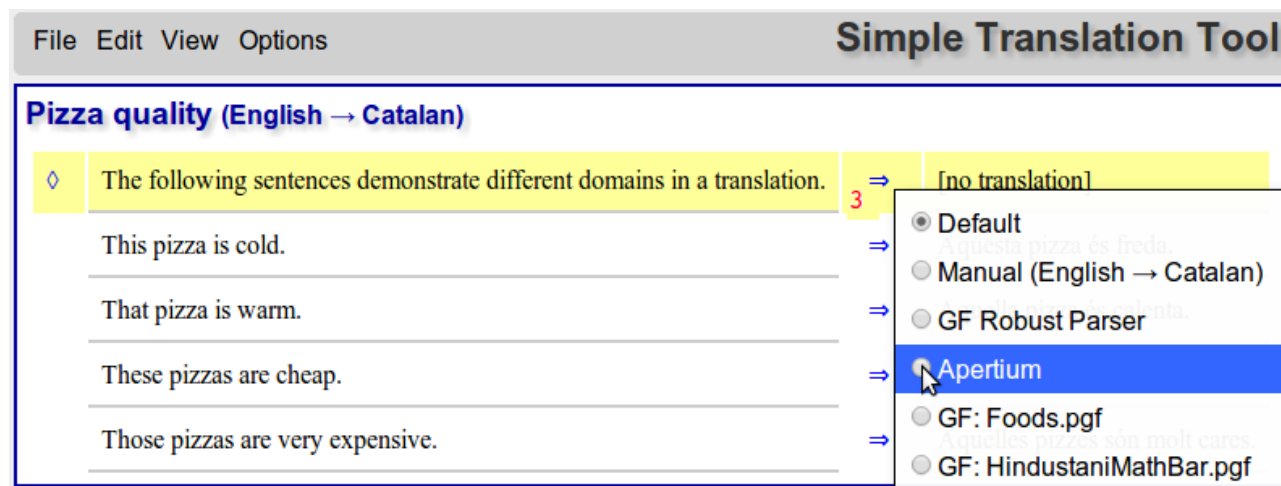
4. Correcting the first error by editing the translation

The second error manifests as no translation. The solution here is to paraphrase the source; in this case, changing the modifier *really* to *very*, that is supported by the lexicon. This process is guided by the Minibar, which is a way to construct grammatical sentences one word at a time. The Syntax Editor is planned to be integrated in the tool, so that the sentences can be modified at a deeper level.



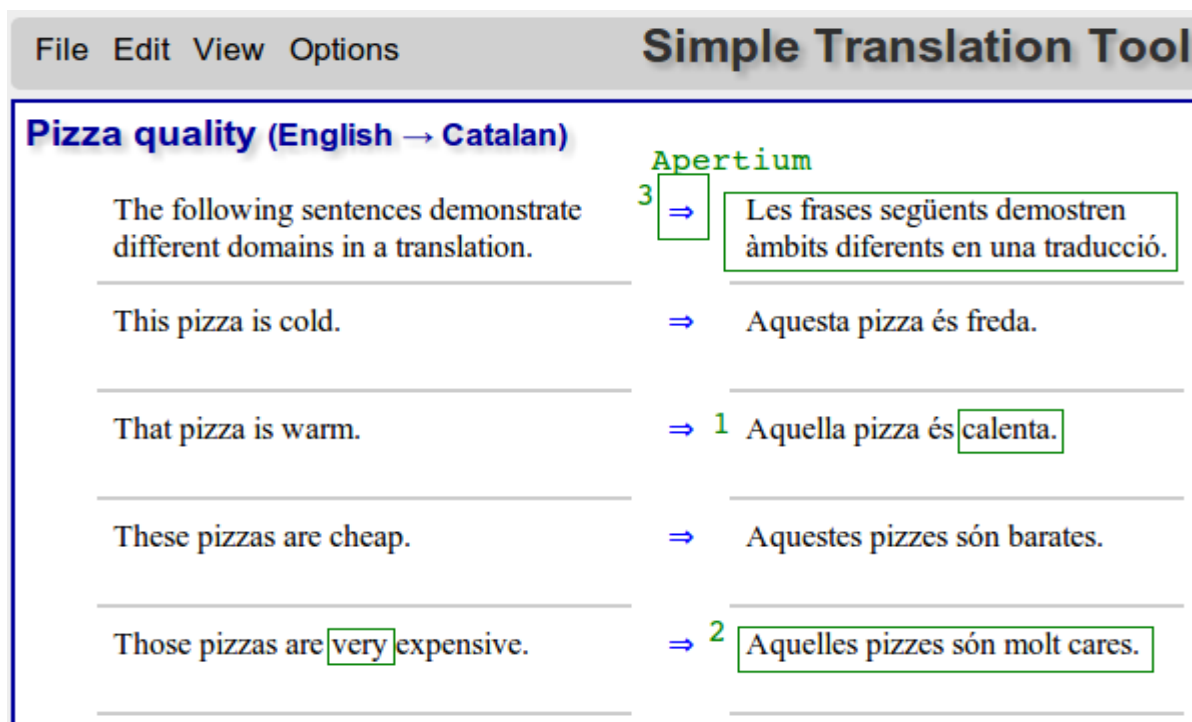
5. Correcting the second error by editing the source

The third error also shows no translation, but it is due to the segment being totally different domain. In the example document of 5 phrases, the first is a commentary, written on completely unrestricted text, and the four remaining phrases are the sort of restricted language that translates with our chosen GF grammar. In a realistic situation, the first phrase could be generic instructions and the latter ones could be mathematical formulas, in order for the scenario to make sense. In any case, the error is corrected by changing the translation method for that segment. Instead of any GF grammar, we choose Apertium, with more coverage but quality not guaranteed. In case the translator spots errors in generic MT option, there is the source post-editing option.

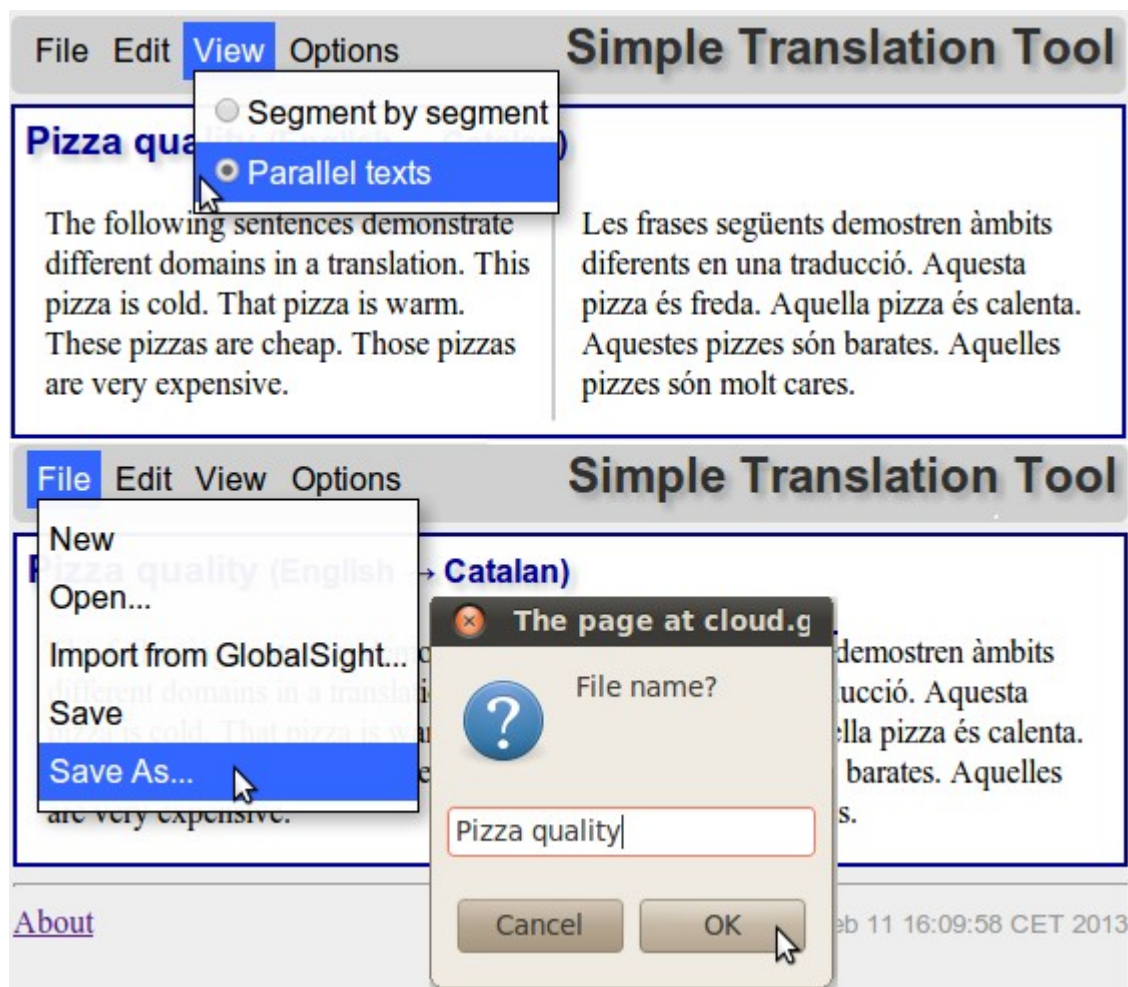


6. Correcting the third error by choosing another translation method

Finally, all three errors have been corrected. The user can view the the document as parallel texts and save the project.



7. All errors are corrected



8. Saving the translation

4. Future work

4.1 Translation memory generation

UHEL will conduct experiments on creating translation memory data with GF grammars. Grammars of a given domain are used to generate bilingual aligned data, which can be converted into a translation memory. Then, when translating new material, the translation memory can provide fuzzy matches for cases where the constructions are similar, but words are different. This is one way to compensate for the lack of lexicon in a situation where adding new vocabulary is hard.

[Jonson](#) (2006) describes an experiment on synthesizing a corpus with GF for training speech recognition models. The idea is similar: use a grammar to generate reliable data for a data-driven approach. By using GF translation suggestions in a translation memory it is also possible to use standard translation tools like Trados to generate pre-translation reports of exact and fuzzy matches. These percentages of different matches are easier to demonstrate to translation industry stakeholders, as the scientific metrics used in MT evaluation (BLEU, NIST, Rouge and so on) are not generally used or well understood within the industry.

4.2 GF cloud

Services in the GF cloud will be linked to each other by UGOT. [Simple Translation Tool](#) will be extended to a bilingual, controlled language document authoring tool, with useful ways to enter and edit the source text too. Additions include a text input guided by word completion and syntax tree editing, by invoking the [Syntax Editor](#) (see Section 2.2) on a source segment.

4.3 Grammar editing

Grammar editing in the translator's tools is still an open question. One of the main shortcomings of a MOLTO type translation is the limited coverage, and that's why it is important that a translator can easily extend and modify the lexicon. Just importing raw lexicon data is described in [Listenmaa](#) (2012). This is a question of adding more content, but another question is modifying existing grammars, usually in a case of an error.

There are some steps taken to further this issue. [Fuchs et al](#) (2013) present a multilingual semantic wiki, where it is possible for every user to modify the grammar behind the wiki. This is still an expert work, as the editing is done with raw GF code, but there are methods for guided grammar editing, such as the cloud-based IDE (see [documentation](#)). This environment offers an easy way of multilingual grammar writing and editing.

References

- Norbert Fuchs, Kaarel Kaljurand, Tobias Kuhn. 2013. Multilingual semantic wiki. *MOLTO Deliverable D11.2*.
- Rebecca Jonson. 2006. Generating statistical language models from interpretation grammars in dialogue system. In *Proceedings of EACL'06*, Trento, Italy.
- Listenmaa, Inari. 2012. *Ontology-based Lexicon Managament in a Multilingual Translation System – a Survey of Use Cases*. M.A. Thesis, University of Helsinki.