PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 93 JANUARY 2010 47–56

Tradubi: Open-Source Social Translation for the Apertium Machine Translation Platform

Víctor M. Sánchez-Cartagena, Juan Antonio Pérez-Ortiz

Dept. Llenguatges i Sistemes Informàtics, Universitat d'Alacant, Spain

Abstract

Massive online collaboration could become a winning strategy to tear down the language barriers on the web, and in order for this to happen appropriate computer tools, like reliable machine translation systems and friendly postediting interfaces, should be widely available. However, community collaboration should not only involve the postediting of machine translations, but also the creation of the linguistic resources needed to improve the translation engines. In this paper we introduce Tradubi, a free/open-source web application for social translation, whose aim is, firstly, to build a platform for collaboratively customising and improving rule-based machine translation systems and, secondly, to offer an environment for the postediting and subsequent sharing of raw machine translations. Currently, Tradubi is built upon the free/open-source Apertium machine translation engine. The application can be accessed at `tradubi.com` or downloaded and installed on a different server.

1. Introduction

The role of internet users has quickly evolved since the irruption of the web in the middle of the nineties: early passive consumers have become active *prosumers* (a word coined to refer to users which are both producers and consumers) of information. Under this view, internet companies *simply* build the spaces for interaction and users *colonise* them. This active role of users constitutes one of the main characteristics of what has been tagged as the *web 2.0* (O'Reilly, 2005).

However, in spite of the vast amount of contents uploaded to the *cloud* (another neologism which is commonly used as a synonym for internet) during the last years, linguistic barriers are still a significant obstacle to universal collaboration since they end up creating *islands* of content, only meaningful to speakers of a particular language.

Corresponding author: japerez@dlsi.ua.es
Cite as: Víctor M. Sánchez-Cartagena, Juan Antonio Pérez-Ortiz. Tradubi: Open-Source Social Translation for the Apertium Machine Translation Platform. The Prague Bulletin of Mathematical Linguistics No. 93, 2010, pp. 47–56. ISBN 978-80-904175-4-0. doi: 10.2478/v10108-010-0012-8.

Massive online collaboration (involving not only professional translators but also amateurs) is probably the only force capable of tearing down these barriers (Garcia, 2009). The resulting scenario, which can be defined as *social translation*, will need efficient computer translation tools, such as machine translation (MT) systems or shared translation memories.

In the particular case of MT, collaboration should not only concern the postediting of raw translations, but also the creation of the linguistic resources needed by MT systems and the improvement of the translation engines. In this paper, we introduce for the first time Tradubi[1], a free/open-source web application whose aim is to ease these steps and become a platform for social translation. At the moment, Tradubi is built upon the Apertium free/open-source platform for rule-based MT (Forcada et al., 2009). With the help of Tradubi, users can create customised dictionaries for Apertium which focus on specific linguistic domains or which correct translation errors made by the default system. Tradubi allows every user or group of users to configure their own Apertium-based machine translator by defining a hierarchy of dictionaries to be used when translating texts.

Besides that, the last version of Tradubi includes a simple mechanism for the storage and management of the postedited translations. This feature is expected to improve in future versions so that users can work collaboratively on the postediting, refinement and publishing of translations.

Section 2 reviews some of the current approaches to social collaboration in the field of translation. Then, section 3 enumerates the main features of the current version of Tradubi. After that, some technical issues related to the development of Tradubi are discussed in section 4. The paper finishes with some conclusions and an overview of the features to be incorporated into future versions of the application.

2. Social Translation on the Web 2.0

There are a lot of web-based services for human translation. A selection of some of the the most relevant follows:

- Cucumis[2] is an online collaborative translation service based on an exchange policy: users gain points when they translate a document and these points are needed if they want to submit a text to be translated by other users (Cucumis' motto is "do you want to translate or to be translated?").
- Traduwiki[3] or Worldwide Lexicon[4] are similar to Cucumis but with a more open policy regarding who can translate or ask for a translation.

[1] Tradubi can be accessed at `http://tradubi.com`, and its source code can be downloaded from `http://tradubi.sourceforge.net`.

[2] `http://www.cucumis.org`

[3] `http://traduwiki.org`

[4] `http://www.worldwidelexicon.org`

- OneHourTranslation[5] is more business-oriented: users pay for translations and the company deducts a small commission from every transaction.

None of the previous sites enforce any particular tool to carry on the translations. A different group of web applications like Wiktionary[6] or Lingro[7] are focused on the collaborative building of dictionaries and terminological databases.

Web-based *tools* for translation can also be found. For example, the recently launched Google Translator Toolkit[8] allows users to create, maintain, use and share translation memories and terminological databases, as well as combining them with statistical MT through a specialised interface for translation inspired on the one popularised by traditional translation memory management systems; users have access to the statistical MT system but they cannot modify directly its behaviour. In connection with the system presented in this paper, that is, one dealing with the configuration of rule-based MT system, some similar approaches can be found in the literature, as, for example, the translation environment *Yakushite Net* (Murata et al., 2003). Our proposal is the first free/open-source and the first to focus on the expanding Apertium platform.

3. Current Features of Tradubi

Tradubi is an Ajax-based (Garret, 2005) web application, that is, an application which can be run in a browser without requiring installation of any additional plugin. The client side (mostly, the interface) of the application is therefore encoded in JavaScript (see 4.2 for more details) and communicates with a server responsible for the tasks which cannot be executed in the browser. This follows an emerging trend on the web where applications are moving from the desktop to the *cloud*.

Users of Tradubi create customised dictionaries of translation units, which consist of a word or sequence of words in the source language and their corresponding translation in the target language (for example, the English–Spanish translation unit *glucose-6-phosphate isomerase/glucosa-6-fosfato isomerasa*). Note that in the current version no morphological information is attached to the words, which makes it easier for non-experts to add new entries but might require to add all the lexical variations of a word in some cases. User dictionaries and translation units can be used in the following ways:

- *Creation:* users can add new translation units to the engine.
- *Maintenance:* the translation units can be modified or deleted at any time.
- *Hierarchy definition:* a group of user dictionaries (for example, for different linguistic domains) can be used in new translations; in order to avoid conflicts, users can define a hierarchy of these dictionaries.

[5] http://www.onehourtranslation.com

[6] http://www.wiktionary.org

[7] http://lingro.com

[8] http://translate.google.com/toolkit/

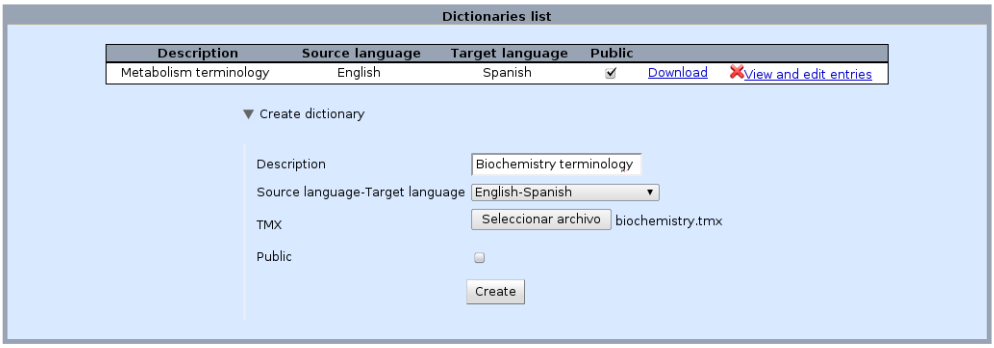

Figure 1. A screenshot of Tradubi showing the creation of a new English-Spanish user dictionary intended for biochemical terms. The dictionary will be initially fed with translation units from the biochemistry.tmx *file. Data will be non-public (private or shareable). The list of current available dictionaries is shown on the top (in this case, a public dictionary with terminology about metabolism).*

- *Sharing:* a user dictionary can be tagged as public, private or shared (in read-only mode at this moment) with other users; when defining a dictionary hierarchy, every available dictionary can be considered.
- *Recommendation:* Tradubi can suggest before translation a dictionary or a set of dictionaries for a particular source text according to the number of words in the text found in the dictionaries.
- *Import/export:* the translation units in a dictionary can be imported or exported using the Translation Memory eXchange[9] (TMX) standard format.
- *Collaborative creation:* shared dictionaries may received translation units from every user with permissions; this feature will allow for the collaborative creation of dictionaries, but it is not implemented in the stable version of Tradubi yet.

Apart from this, postedited translations can be stored and retrieved at any time; this feature will evolve to a system for collaborative postediting and dissemination of translations.

Figures 1 to 3 show some screenshots of the application with some additional comments.

[9] `http://www.lisa.org/standards/tmx/`

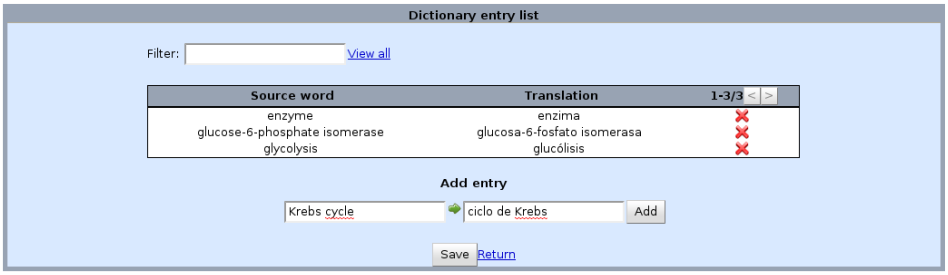

*Figure 2. A screenshot of Tradubi showing the addition of a new translation unit (*Krebs cycle/ciclo de Krebs*). The dictionary already contains three translation units which can be modified or deleted. The* save *button triggers the compilation of the dictionary to the binary form used by Apertium.*

4. Technical Issues

Tradubi's design, development and deployment requires dealing with a number of technical issues which are discussed in this section.

4.1. License Choice

Tradubi is not only available through a public web server, but also as a free/open-source program which can be downloaded, installed and modified by everyone. It is licensed under version 3 of the GNU Affero General Public License[10] (AGPL). This license is fully compatible with the GNU General Public License (GPL) and equally proposed by the Free Software Foundation, which in fact recommends[11] that "developers consider using the GNU AGPL for any software which will commonly be run over a network". AGPL has been suggested as a means to close a *loophole* in the ordinary GPL which does not force organisations to distribute derivative code when it is only deployed as a web service.

Choosing AGPL is a little big controversial (O'Grady, 2009) since this license adds a new constraint to the well-established GPL license. We consider, however, that in the web 2.0 era and with the traditional model of software distribution gradually losing ground to the cloud computing model, AGPL should be being adopted by a higher number of free/open-source projects.

[10] `http://www.gnu.org/licenses/agpl-3.0.html`

[11] `http://www.fsf.org/licensing/licenses/`

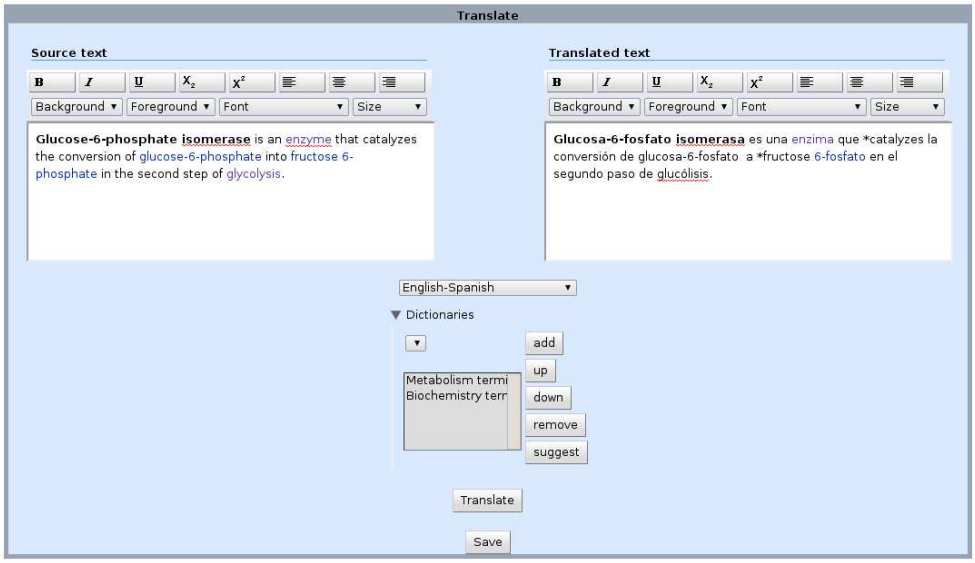

Figure 3. A screenshot of Tradubi showing a translation with two user dictionaries (one on metabolism, with higher precedence, and a second one on biochemistry). The two dictionaries include a different translation for the word glycolysis *(*glucólisis *and* glicolisis*) but the one in the first dictionary (*glucólisis*) has been chosen because of the hierarchy defined in this case. Apart from this explicit choice of dictionaries, the* suggest *button automatically selects the most appropriate dictionary according to the source text. The resulting machine translated text on the right is ready to be postedited and then saved.*

4.2. Programming Language and Framework

Tradubi client is mostly written in Java with the help of the Google Web Toolkit[12] (GWT). GWT is a free/open-source framework for developing web applications. At the core of the framework is a compiler which translates the code written for the client in Java to JavaScript code which runs flawlessly in current browsers. GWT simplifies the development and debugging of Ajax-based web applications which require asynchronous remote procedure calls, history management, bookmarking, internationalisation or code splitting.

[12] `http://code.google.com/webtoolkit/`

The code for the server side is written in Java as well. The project code consists (as of current version) of 135 classes and around 13 000 lines of code.

4.3. Data Portability and Accessibility

According to the DataPortability Project,[13] *data portability* is the "ability for people to reuse their data across interoperable applications". This is key feature which the web 2.0 should embrace in order to mitigate the undesirable consequences of *walled gardens*. Import and export of the dictionaries in TMX format allow Tradubi users to seamlessly move their data, for example, to Google Translator Toolkit.[14]

In addition to this, Tradubi users may log into the application using an existing OpenID account.[15] User interface and dynamic content are more accessible thanks to the adoption of the WAI-ARIA[16] standard.

4.4. Apertium Server

Communication between Tradubi and the Apertium engine is done via an already implemented scalable architecture (Sánchez-Cartagena and Pérez-Ortiz, 2009) for Apertium. This architecture consists of a *router server* which forwards incoming translation requests to one or more slaves running Apertium instances. Our web application sends requests to the router through the Remote Method Invocation (RMI) protocol; although a convenient Application Programming Interface (API) is also available, we chose to use RMI directly since Tradubi and the scalable translation system are both written in Java.

4.5. Integration with Apertium

As already commented, Tradubi allows every user or group of users to configure their own Apertium-based machine translator by defining a hierarchy of dictionaries to be used when translating texts: matched entries in a dictionary at level i of the hierarchy take precedence over any other match found in a dictionary at level j with $i < j$, default system dictionaries being at the highest level (that is, they have the minimum precedence).

User dictionaries are specified, compiled and accessed as regular Apertium monolingual dictionaries (Forcada et al., 2009), except for the fact that no morphological information is attached to the entries. Originally, every translation unit is coded in

[13] `http://www.dataportability.org`

[14] Note that, at the time of writing, Google Translator Toolkit does not allow users to export their own data.

[15] `http://openid.net`

[16] `http://www.w3.org/WAI/intro/aria`

XML inside an *e* element containing the source and target words; the following is an excerpt of an English–Spanish user dictionary:

```
<e><p>
  <l>glucose-6-phosphate<b/>isomerase</l>
  <r>glucosa-6-fosfato isomerasa</r>
</p></e>
```

The XML file is then compiled to a binary form by means of the *lttoolbox* library included in Apertium. The binary version of the dictionaries implement a finite-state transducer (Roche and Schabes, 1997) which is used to efficiently detect and translate the source words. This compilation is done as soon as the user clicks on the *save* button after introducing or modifying a set of translation units (see figure 2); therefore, the new units are ready immediately for new translations.

The resulting transducer is inserted in the Apertium pipeline between the part-of-speech tagger and the structural transfer module. This way, the tagger has more information for disambiguation since it can consider the lexical categories of words in the default dictionaries of Apertium which, however, are going to be translated with a user dictionary. If a user defines a hierarchy of dictionaries the system is set up as a cascade of modules that successively search for the words in the source text and keep the first translation found.

Some problems arise when a translation unit contains a word that is part of a *multiword* in the default system dictionaries. For example, if default dictionaries contain an entry for *an arm and a leg* and a user dictionary contains the translation unit *leg/etapa*, then the word *etapa* will never appear in the target text when translating a source sentence which includes the multiword.

4.6. Compilation and Installation

Source code can be downloaded from the SVN repository of Tradubi located at *Sourceforge.net*.[17] It includes documentation with additional instructions on how to compile and install the application.

5. Future Work on Tradubi

Tradubi is still in its early stages of development, but with some of the following improvements we expect it to become a mature and stable framework for social translation.

It is worth studying alternative places of insertion into the Apertium pipeline of the new modules dealing with user dictionaries. Currently they are located just before the

[17] `http://tradubi.sourceforge.net`

structural transfer module, but locating them in other positions (for example, before the morphological analyser) could result in the overcoming of the multiword problem (see 4.5) while keeping functionality the same.

We also plan to consider the inclusion of other MT engines in addition to Apertium. The open-source nature of Apertium has allowed us to easily insert the new modules for user dictionaries in the middle of its pipeline, but it might be very interesting to research how to extend these modules to engines with no source code available and which can be accessed online through web services only. This would require to consider how to isolate the words found in the user dictionaries from the rest of the text which should be translated by the MT engine.

In harmony with the idea of adopting the principles of open data (see 4.3), we will also implement an option for downloading for local installation a package with the Apertium engine and all the linguistic data and user dictionaries making up a particular translator that a user has configured online.

The current simple interface for postediting will evolve into a more friendly interface which benefits from information extracted from the MT engine in a way similar to recent proposals in the statistical MT field (Koehn, 2009). The information collected from the interaction of the users with the postediting interface will also be used to improve the linguistic data (both dictionaries and structural transfer rules) of the Apertium-based translators in a similar manner to the *Translation Correction Tool* (Font-Llitjós et al., 2005).

Finally, more social features could be added to the application.

6. Conclusions

This is the first paper to introduce Tradubi, a free/open-source Ajax-based web application for the collaborative configuration of rule-based MT systems. Currently, Tradubi works as a layer over Apertium, allowing users to create and use hierarchies of dictionaries which override default system dictionaries in case of conflict. We expect to augment the functionalities of Tradubi so that it becomes a powerful application for social translation.

7. Acknowledgements

This work has been partially funded by Spanish Ministerio de Ciencia e Innovación through project TIN2009-14009-C02-01.

Bibliography

Font-Llitjós, Ariadna, Jaime Carbonell, and Alon Lavie. A framework for interactive and automatic refinement of transfer-based machine translation. In *Proceedings of EAMT 10th Annual Conference*, 2005.

Forcada, Mikel L., Francis M. Tyers, and Gema Ramírez-Sánchez. The Apertium machine translation platform: five years on. In *Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation*, pages 3–10, 2009.

Garcia, Ignacio. Beyond translation memory: Computers and the professional. *The Journal of Specialised Translation*, 12:199–214, 2009.

Garret, Jesse James. Ajax: A new approach to web applications. *AdaptivePath.com,* `http://www.adaptivepath.com/ideas/essays/archives/000385.php`, 2005.

Koehn, Philipp. A Web-Based interactive computer aided translation tool. In *Proceedings of the ACL-IJCNLP 2009, Software Demonstrations*, pages 17–20, 2009.

Murata, T., M. Kitamura, T. Fukui, and T. Sukehiro. Implementation of collaborative translation environment: YakushiteNet. In *Proceedings of MT Summit IX*, 2003.

O'Grady, Stephen. AGPL: Open source licensing in a networked age. *RedMonk.com,* `http://redmonk.com/sograady/2009/04/15/open-source-licensing-in-a-networked-age/`, 2009.

O'Reilly, Tim. What is web 2.0. *O'Reilly Network,* `http://oreilly.com/web2/archive/what-is-web-20.html`, 2005.

Roche, Emmanuel and Yves Schabes. *Finite-state language processing*. MIT Press, 1997.

Sánchez-Cartagena, Víctor M. and Juan Antonio Pérez-Ortiz. An open-source highly scalable web service architecture for the Apertium machine translation engine. In *Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation*, pages 51–58, 2009.