When Translation Requires Interpretation: Collaborative Computer–Assisted Translation of Ancient Texts

D. Albanesi[1], **A. Bellandi**[1], **G. Benotto**[1], **G. Di Segni**[2], **E. Giovannetti**[1]
[1] Istituto di Linguistica Computazionale, Consiglio Nazionale delle Ricerche
Via G. Moruzzi 1, 56124, Pisa - Italy
name.surname@ilc.cnr.it
[2] Istituto di Biologia Cellulare e Neurobiologia, Consiglio Nazionale delle Ricerche
Via Ramarini 32, 00015, Monterotondo (Rome) - Italy
gianfranco.disegni@cnr.it

Abstract

This paper introduces the main features of *Traduco*, a Web-based, collaborative Computer-Assisted Translation (CAT) tool developed to support the translation of ancient texts. In addition to the standard components offered by traditional CAT tools, *Traduco* includes a number of features designed to ease the translation of ancient texts, such as the Babylonian Talmud, posing specific structural, stylistic, linguistic and hermeneutical challenges.

1 Introduction

We here describe *Traduco*, a collaborative Web application designed to support the translation of ancient texts, developed in the context of a project for the translation of the Babylonian Talmud (BT) into Italian. *Traduco* extends most of the standard components of a traditional Computer–Assisted Translation (CAT) tool with specific features needed to support the translation of ancient texts such as the BT. The design and development of *Traduco* required the adoption of a multidisciplinary approach, leveraging on advances in software engineering, knowledge engineering, computational linguistics, Talmudic knowledge, Semitic linguistics and publishing. The Babylonian Talmud consists in the teachings of the Masters of Judaism in a span of six centuries, until the fifth century, and it is divided into Mishnah and Gemara. The Babylonian Talmud consists of 5422 pages. It is not a unified work but it is a collection of sayings of many different Masters, delivered in the course of several generations, partly in Hebrew and mostly in Aramaic. It has a complex layering and it is written in a concise manner, difficult to understand, using many idioms that, if translated literally, would be incomprehensible. The way in which the discussion develops is that of questions and answers, objections and attempts to reply. Many passages occur in different tractates, with or without variants. Having dealt with the translation of a literary creation of such hermeneutical complexity, richness and heterogeneity of topics as the Babylonian Talmud, *Traduco* can be easily applied to support the translation of other ancient texts and to manage other languages.

2 The *Traduco* System

Computer–Assisted Translation (CAT) tools are designed to aid in the translation of a text (Christensen and Schjoldager, 1996; Gordon, 1996; Planas, 2005; Barracchina et. al, 2009). The core technology of a CAT tool is the Translation Memory (TM), a repository that allows translators to consult and reuse past translations, primarily developed to speed up the translation process (Reinke, 2006; Somers, 2003; Koehn, 2009; O'Brien, 2006; Planas and Furuse, 1999). However, considering the nature of the texts we are working on (as the BT), the quality of the translation is much more important than the translation pace. For this reason, a system developed to support the translation of ancient texts must go beyond the standard set of functionalities offered by a traditional CAT tool. Moreover, particularly complex texts, as the BT, can require the competencies of a multiplicity of specialized users that must be able to translate the very same text in a collaborative way on a Web environment. The most used available open source CAT tools (OpenTM[1], OmegaT[2], Transolution[3], Olanto[4], MateCat[5], MASMCAT[6])

[1] http://www.opentm2.org/
[2] http://www.omegat.org/
[3] http://www.tran-solution.net/
[4] http://olanto.org/
[5] https://www.matecat.com/
[6] http://www.casmacat.eu/

Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, pages 84–88,
Beijing, China, July 30, 2015.

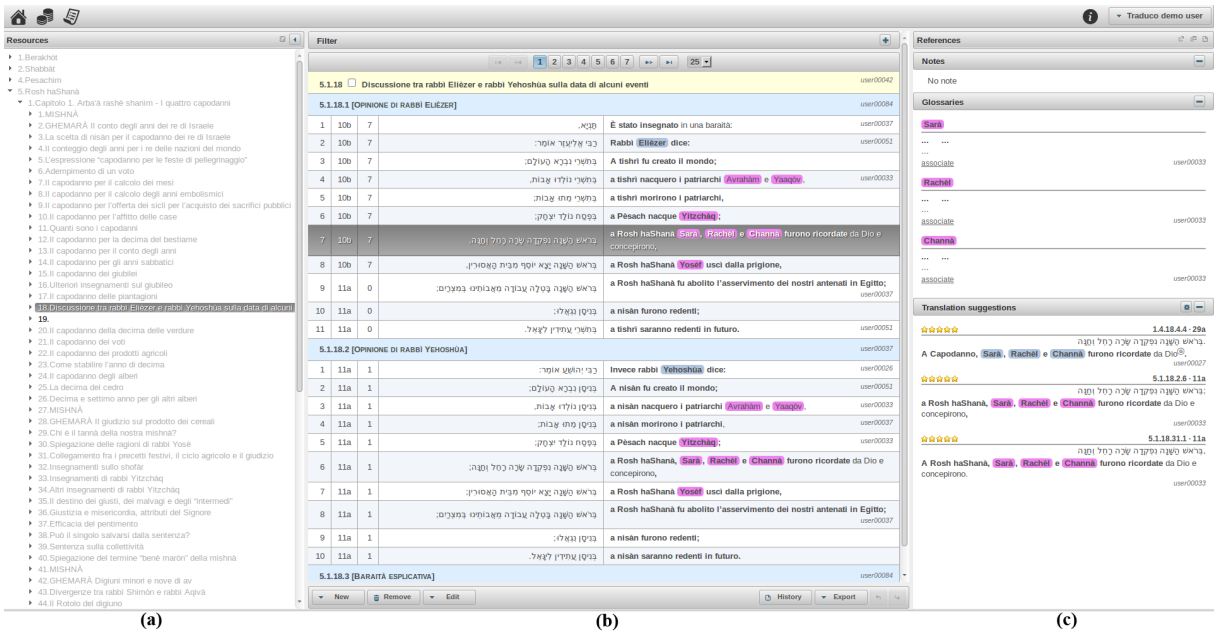

Figure 1: The main GUI of *Traduco* system. (a) hierarchical structure of the translated text; (b) translation table; (c) translation references: notes, glossaries, translation suggestions.

and even the main commercial tools (Deja Vu[7], SDL Trados[8], memoQ[9], memsource[10]) are not suitable for the collaborative translation of ancient texts (Bellandi, 2014), since they do not respond to the specific needs of those specialists. In the following, we briefly illustrate some of *Traduco*'s features specifically designed to answer to these needs. We strongly encourage to try the demo version of *Traduco*[11] (all the references have been blocked for the review process).

Manual segmentation process and hierarchical structuring.
Typically, a CAT tool automatically segments the source text into sentences. However, several different languages and dialectal variants alternate in the text of the BT, making difficult, if not impossible, to develop an automatic (statistical or pattern–based) tool able to split sentences on language transitions. In addition, to maximise the use of the TM and to reuse past translations, it is necessary to isolate the formulaic expressions scattered all over the whole text, even if they do not cover an entire sentence. *Traduco* eases the process of manual segmentation by providing the "Generate" function. Instead of translating pericope by pericope[12], a translator can insert, at once, a sequence of pericopes: a whole portion of text can be pasted inside a specific text field and split into distinct lines, each of which will be interpreted as a single pericope. To ease the translation process a rich text editor is provided with a series of buttons opening subpanels (see Figure 2). From left to right: bold (to indicate literal translations), italic (to indicate quotations from the Bible), underline (to mark Hebrew words for publishing purposes), small caps (for the Mishnah and quotations from the Mishnah), notes, semantic annotations, undo, redo, remove formatting, show HTML source, special characters (e.g. for transliterations, quotation marks, etc.), and, finally, six shortcuts for bibliographic references (e.g., Bible, Legal Code, Mishnah, etc.). Furthermore, due to the complexity of the inner structure of the BT, *Traduco* allows to hierarchically organize the translation both to preserve the structure of the source text (e.g. in the

[7] http://www.translation.net/deja_vu_x.html
[8] http://www.translationzone.com/products/sdl-trados-studio/
[9] https://www.memoq.com/get-memoq
[10] https://www.memsource.com/en
[11] Test *Traduco* at http://146.48.92.138:8082/talmud (user: traducodemo - password: traducodemo): we recommend to use Mozilla Firefox. You will also find an exhaustive use-cases guide to test the main features of the System (click on the "i" button, once logged in). Parts of data, authors, and parts of the original BT text have been clouded for privacy and rights reasons.
[12] A pericope defines a portion of text having an arbitrary length.

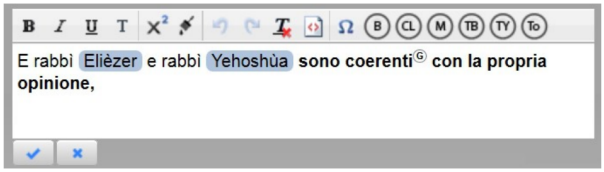

Figure 2: The rich text editor. The sentence means: "Rabbi Eliézer and Rabbi Yehoshúa are coherent with their own opinion"

case of the BT, tractates, chapters, and blocks[13]) and to add customized levels (e.g. logical units[14]).

Literal and explicative translations, notes.
An ancient writing such as the BT cannot be translated as a modern text, since a literal translation would not be intelligible to a modern reader. Therefore, a good translation of the BT requires the addition of explicative integrations within the translation, which is not merely a translation, but, to a certain extent, an interpretative commentary in itself. To do so, *Traduco* provides tools that enable to distinguish the literal part of the translation (indicated in bold) from the explicative additions of the translators/scholars. Furthermore, it allows the insertion of different types of explicative notes in the text (see Figure 2 for an example of a generic note, shown as a little "G" inside a circle, following the word "coerenti": the text of the note will be inserted in a dedicated panel in the upper right part of the interface).

Revision support: multiple user roles, versioning.
Traduco offers a multirole environment: users can either be translators, revisors, editors or administrators. Concerning revisors, they can edit translations done by translators, which, in turn, can exploit the versioning system to keep track of the history of each resource (translated pericope, note, glossary entry). Additionally, revisors can bring the translators' attention to a specific portion of translation by adding special revision notes. Finally, revisors and editors can work together to attain a more coherent, homogeneous and fluent translation of the text by comparing each translation to the ones the suggestion component shows.

[13] A block corresponds to a discussion about a homogeneous and well-defined subject.

[14] A logical unit is a part of a block with a defined logic, e.g. thesis, hypothesis, objection, question, biblical quotation, etc.

Ranking of translation suggestions.
Being a collaborative Web environment, *Traduco* can rank the translation suggestions (Vanallemeersch, 2015; Wolff, 2014) stored in the TM on the basis of several parameters, including the authoritativeness of the translator that produced the suggested translation and the tractate the suggestions belong to.

Semantic annotation and glossaries.
Since the BT translation includes discussions regarding many different fields of knowledge (jurisprudence, liturgy, ethics, rituals, philosophy, trade, medicine, astronomy, etc.), it can greatly benefit from semantic annotations, in order to provide readers with further assistance in the interpretation of the text. For the translation of the BT, *Traduco* provides a set of six predefined semantic classes: concepts, linguistic expressions, Rabbis, measures, nature, and persons. This functionality allows the creation of specialized glossaries that can be queried and browsed in a dedicated section of the system. Annotations can be done by selecting the text and then choosing one of the classes in the sub–panel opening through the paintbrush button of the editor (see Figure 2). A semantic annotation is represented with a specific colored highlighting: in Figure 2, for example, two names of Rabbis have been annotated and highlighted in gray. Each annotation can be accompanied by a free textual description (see the "Glossaries" panel on the right of Figure 1(c)), an optional transliteration and an optional Hebrew original form. A new annotation can be associated to a canonical form by referring to an existing glossary entry: it can be done with the "Associate" link at the bottom of the "Glossaries" panel. Furthermore, glossary entries can be browsed and searched in a dedicated interface, opened via the "Glossaries" button on the upper left part of the main GUI.

3 General Architecture and Technical Solutions

From the technical point of view, *Traduco* was designed as a group of independent web–based components connected by interfaces. It is based on the software design pattern known as "three–tier architecture", and it exploits Apache Tomcat v7.0 as web server. The system was implemented using

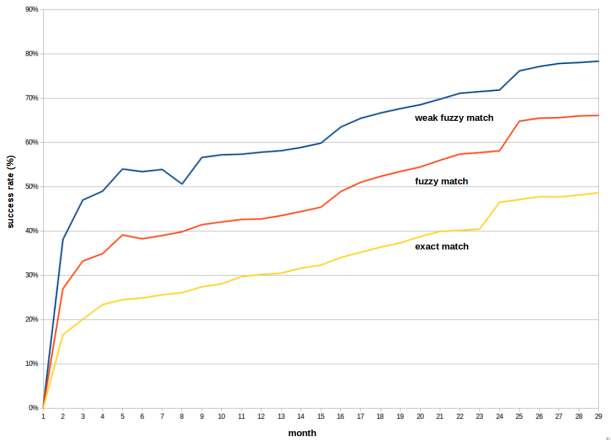

Figure 3: The TM redundancy curves w.r.t. the ranking of the similarity function ED.

the Java 2 Standard Edition (J2SE) framework, allowing, among the other things, to easily manipulate unicode characters and thus to manage other languages. Relational persistence and query services are managed by Hibernate v4.3.7 that takes care of the mapping from Java classes to the Mysql v5.0 database tables. The presentation tier has been implemented by means of JavaServer Faces (JSF). As JSF library, we used Primefaces v5.1. To accomplish the translation suggestion task, the system includes a Translation Memory (TM) designed to remember every translated portion of text, organized at the pericope level. For each pericope, the TM contains the translation, the author of the translation, and the reference to the tractate to which the pericope belongs (here called context). In order to develop a language independent component, we took account of adopting similarity measures based on edit distance, $ED(p_1, p_2)$, by considering two pericopes to be more similar when the same terms tend to appear in the same order. Given a pericope p_q of length $|p_q|$, and a distance error δ, our similarity function allows to both retrieve all pericopes in the TM (called suggestions) such that $ED(p_q, p) \leq round(|p_q|)$, and rank suggestions, not only on the basis of the ED outcome, but also on both the current context and the suggestion author. In order to take into account the lenght of the query (p_q), we considered δ as the percentage of admitted errors w.r.t. the sentences to be translated, multiplying it by the length of the query segment (Mandreoli et al., 2002). In collaboration with the translators' team, we have experimentally tuned δ to 0.7. Our similarity algorithm is based on dynamic programming, and its implementation refers to (Navarro, 2001). The inverted index data structure is the central component of our search engine indexing algorithm, for accessing the TM. The goal of our search engine implementation is to optimize the speed of the queries to provide a more efficient suggestion of the pre-existing translations. In particular, we used a record level inverted index technique, containing a list of references to pericopes for each word. In order to roughly estimate the degree of redundancy of the TM, we conducted a jackknife experiment (Wu, 2009), as reported in Figure 3. The curve labelled with "exact match" represents perfect suggestions, while the one labelled with "fuzzy match" indicates that few corrections are required to improve the suggestion. Finally, the curve labelled with "weak fuzzy match" refers, in most of the cases, to acceptable suggestions. The percentage of source segments found both verbatim and fuzzy in the memory grows logarithmically both with time and the size of the TM.

4 Conclusions and Perspectives

It is renowned that CAT techniques work best on texts that are highly repetitive, and for this reason they are mainly applied to the translation of technical manuals. They are also helpful for translating incremental changes in a previously translated document, corresponding, for example, to minor changes in a new version of a user manual. Thus, Translation Memories have not been considered appropriate for literary or creative texts. One of the novelty of our work is that of applying this kind of approach to the process of translating ancient texts. In general, these texts share common features, both from the content and the linguistic perspective. As exemplified in particular by our test bed, ancient texts can be lexically poor and repetitive by nature, they have a complex inner structure that has to be taken into account while translating, and they necessarily need annotations at various levels of granularity in order to make ancient concepts expressed in the texts understandable to contemporary readers. Such complexity also entails that, in order to be properly translated, these texts should be processed by a team of scholars with heterogeneous competences. Our system satisfy this need by introducing the idea of collaborative work to CAT and by enhancing it with tools apt to satisfy the different users competences (i.e., translators, revisors, editors).

Acknowledgments

This work has been conducted in the context of the research project TALMUD and the scientific partnership between S.c.a r.l. "Progetto Traduzione del Talmud Babilonese" and ILC-CNR and on the basis of the regulations stated in the "Protocollo dIntesa" (memorandum of understanding) between the Italian Presidency of the Council of Ministers, the Italian Ministry of Education, Universities and Research, the Union of Italian Jewish Communities, the Italian Rabbinical College and the Italian National Research Council (21/01/2011).

References

Sergio Barracchina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio Lagarda, Hermann Ney, Jesús Tomás and Enrique Vidal. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1): 3-28.

Andrea Bellandi, Alessia Bellusci, Emiliano Giovannetti. 2014. *Computer Assisted Translation of Ancient Texts: the Babylonian Talmud Case Study*. In: Proceedings of the 11th International Natural Language Processing and Cognitive Systems, pp. 287-302, Venice, Italy.

Tina P. Christensen and Anne Schjoldager. 1996. Translation-memory (TM) research: what do we know and how do we know it? *Hermes, Journal of Language and Communication Studies*, 44: 89-101.

Ian Gordon. 1996. *Letting the CAT out of the bag–or was it MT?* In: Proceedings of the 8th International Conference on Translating and the Computer, Aslib, London.

Philipp Koehn. 2009. *A web-based interactive computer aided translation tool*. In: Proceedings of the ACL-IJCNLP 2009 Software Demonstrations. Association for Computational Linguistics, 2009. p. 17-20.

Federica Mandreoli, Riccardo Martoglia, and Paolo Tiberio. 2002. *Searching Similar (Sub) Sentences for Example-Based Machine Translation*. In: Atti del Decimo Convegno Nazionale su Sistemi Evoluti per Basi di Dati (SEBD). Portoferraio (Isola dÉlba), Italy.

Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1): 31-88.

Sharon O'Brien. 2006. Eye-tracking and translation memory matches. *Perspectives: Studies in translatology*, 14(3): 185-205.

Emmanuel Planas, Osamu Furuse. 1999. *Formalizing translation memories*. In: Machine Translation Summit VII. 1999. p. 331-339.

Emmanuel Planas. 2005. *SIMILIS Second-generation translation memory software* In: Proceedings of the 27th International Conference on Translating and the Computer, Aslib, London.

Uwe Reinke. 2006. Translation Memories *Encyclopedia of Language and Linguistics*, 61-65.

Harold L. Somers. 2003. Translation memory systems. *Benjamins Translation Library*, 35 (2003): 31-48.

Tom Vanallemeersch, Vincent Vandeghinste. 2015. *Assessing Linguistically Aware Fuzzy Matching in Translation Memories.* In: Proceedings of the 18th Annual Conference of the European Association for Machine Translation (EAMT2015), Antalya, Turkey.

Friedel Wolff, Laurette Pretorius, Paul Buitelaar. 2014. *Missed Opportunities in Translation Memory Matching.* In: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC): 4401-4406, Reykjavik, Iceland.

Jeff Chien-Fu Wu. 1986. Bootstrap and Other Resampling Methods in Regression Analysis. *The Annals of Statistics*, 14(4): 1261-1295.