# Can Real-Time Machine Translation Overcome Language Barriers in Distributed Requirements Engineering?

Fabio Calefato, Filippo Lanubile, Pasquale Minervini
Dipartimento di Informatica
Università degli Studi di Bari
Bari, Italy
calefato,lanubile@di.uniba.it, pasquale.minervini@uniba.it

***Abstract*—In global software projects work takes place over long distances, meaning that communication will often involve distant cultures with different languages and communication styles that, in turn, exacerbate communication problems. However, being aware of cultural distance is not sufficient to overcome many of the barriers that language differences bring in the way of global project success. In this paper, we investigate the adoption of automatic machine translation (MT) services in synchronous text-based chat in order to overcome any language barrier existing among groups of stakeholders who are remotely negotiating software requirements. We report our findings from a simulated study that compared the performance and the effectiveness of two MT services in translating the messages exchanged during four distributed requirements engineering workshops. The results show that (a) Google Translate produces significantly more intelligible translations than Apertium from English to Italian; (b) both services can be used in text-based chat without disrupting real-time interaction.**

***Keywords-machine translation; cultural distance; language barrier; distributed development; requirements engineering; simulation***

## I. INTRODUCTION

Global software development requires close cooperation of individuals with distant cultural backgrounds. Cultural distance stems from the degree of difference between the sites and manifests itself in two forms, organizational culture and national culture. Organizational culture encompasses the unit's norms and values, including methodologies such as project management practices [7]. More interesting to our research, national culture encompasses an ethnic group's norms, values, and spoken language, often delineated by national boundaries [7].

Cultural difference poses formidable challenges for achieving a shared understanding of the requirements, especially due to language disparities between stakeholders involved [17]. Language is an important component of national cultural distance and a factor that largely accounts for the success of offshore IT work in countries with strong English language capabilities, such as Ireland, the Philippines, and Singapore [7]. Indeed, when language difficulties begin to cause confusion, cultural differences can worsen awkward situations [13]. However, being aware of cultural distance is not sufficient to overcome many of the barriers that language differences bring in the way of global project success [20].

In this paper, we investigate the adoption of automatic machine translation (MT) services in a synchronous text-based chat in order to prevail over language barriers when stakeholders are remotely negotiating software requirements. We selected requirements engineering as the appropriate domain for this study because it is the most communication-intensive activity of software development and thus the one that is alleged to suffer more from language difficulties.

Considering the rather exploratory nature of this study, we run a simulation in which we used two MT systems, namely Google Translate[1] and Apertium[2], to translate the logs collected from requirements engineering workshops. The remainder of this paper is structured as follows. In Section II we briefly overview the machine translation research field, showing the two approaches and systems used in our simulation. Section III goes onto describing our simulation procedure run in order to explore the performance of MT in real-time text chat. The findings from our simulation are presented and discussed, respectively, in Section IV and Section V. Finally, we conclude in Section VI.

## II. MACHINE TRANSLATION

Machine translation (MT) may be defined as the use of a computer to translate a text from one natural language, the source language, into another one, the target language [31]. MT is difficult mainly because translation *per se* involves a huge amount of human knowledge that must be coded in a usable form. Natural languages are highly ambiguous; two languages do not always express the same content in the same way [3]. Research in the field has defined several approaches to develop MT systems, such as the rule-based and the corpus-based approaches.

[1] http://translate.google.com
[2] http://www.apertium.org

### A. Rule-based approach to MT: Apertium

Rule-based MT systems use knowledge in the form of rules, explicitly coded by human experts, which attempt to codify the translation process. Rule-based systems heavily depend on linguistic knowledge, such as bilingual dictionaries [3]. The most notable advantages of rule-based MT services include: more accurate translations, that is, translations more faithful to the meaning of the original text; the ability to explicitly encode linguistic knowledge so that both humans and automatic systems can process; the ease of diagnosing and fix translation errors, like wrong rules in modules or wrong entries in dictionaries. Nevertheless, a rule-based approach has its drawbacks too; most notably a considerable human effort is required in order to develop the necessary linguistic resources (e.g. vocabularies and grammars).

Apertium is an open source, rule-based machine translation platform, which provides SOAP and REST interfaces to the translation service [2]. As of this writing, it supports the translation between any two pairs of over 30 languages.

In [22] we measured the performance of the Apertium service. The performance was evaluated in terms of efficiency (i.e. the time taken to perform translations of sentences of growing length) and scalability (i.e. the time taken to perform translations requested by a growing number of concurrent clients). In our previous work, however, we did not address the quality of the translation provided by Apertium.

### B. Corpus-based approach to MT: Google Translate

Corpus-based MT systems use large collections of parallel texts (i.e. pairs consisting of a text in a source language and its translation into a target language) as the source of knowledge from which the engine learns how to perform translations. Corpus-based MT systems tend to produce translations more fluent than rule-based systems, which instead appear to be more "mechanical". However, such approach requires large amounts of parallel texts (in the order of tens of millions of words) to achieve reasonable translation quality [26]. Compared to the rule-based approach, the corpus-based approach is particularly appealing to researchers because systems can be trained automatically, without any direct human intervention.

Google Translate is an example of statistical MT system that follows the corpus-based approach. In fact, the system does not apply grammatical rules, since its algorithms are based on statistical analysis rather than traditional rule-based analysis. Instead, Google Translate applies statistical learning techniques to build a translation model, relying on a large number of words of text, both monolingual text in the target language and text consisting of examples of human translations between the source and the target languages.

The Google Translate service can be used by third-party applications because it exposes a RESTful interface that returns responses encoded as JSON results. As of this writing, Google Translate supports the translation between any two pairs of over fifty languages.

### C. A MT plugin for eConference

eConference [6] is a text-based distributed meeting system. The primary functionality provided by the tool is a closed group chat, augmented with agenda, meeting minutes editing, and typing awareness capabilities. The tool is built on Eclipse RCP, a pure-plugin platform that allows for full extensibility.

We developed a plugin for eConference that allows selecting both the MT service and the language pair to employ for automatically translating incoming messages during one-to-one and group chat sessions. When a new message is processed by eConference, the MT plugin invokes the configured MT service using the proper web-service interfaces, in order to show the translated messages along with the original text.

Figure 1 shows a screenshot of eConference, with the MT plugin installed, and an example of a one-to-one chat session real-time translation, using the Apertium service (Figure 1a), with original sentence written in English (Figure 1b) translated to Italian in box (Figure 1c).

## III. RELATED WORK

Machine translation is an established technology, some 50 years in the making. The technology available today – i.e. real-time, online conversation – is experiencing tremendous growth of interest, on the heels of the Internet continuous expansion.

As business becomes more global and firms open offices in other countries, the need for companies to communicate in multiple languages with customers, partners, and employees becomes increasingly important [30]. These trends have increased the demand for computer-based translation technology research. In [14] Hogan & Frederking presented WebDIPLOMAT, a MT service that aims to produce more accurate translation by building a statistical model from the combination of multiple MT services already available. In [4] Bangalore *et al.* evaluated the translation quality of a MT service trained using a text corpus made of chat logs collected from the Hubbub prototype used by AT&T employees. Translation quality was measured in terms of the changes (i.e. moves, corrections, and substitution of words) necessary to turn the MT output into the chosen reference translation. However, the approach of choosing *a priori* a reference translation as the correct one has a major drawback in the sense that many correct translations of the same input sentence may exist, despite being completely different in terms of style. Yamashita *et al.* [24][25] studied the effects of machine translation on mutual understanding, which is affected by the asymmetry of machine translation since the sender of a message does not know how well it has been translated to the target language. A limitation of this study is that the researches employed picture description as the experimental tasks, thus focusing mostly on the difficulties arising when describing objects in machine translated discussions.

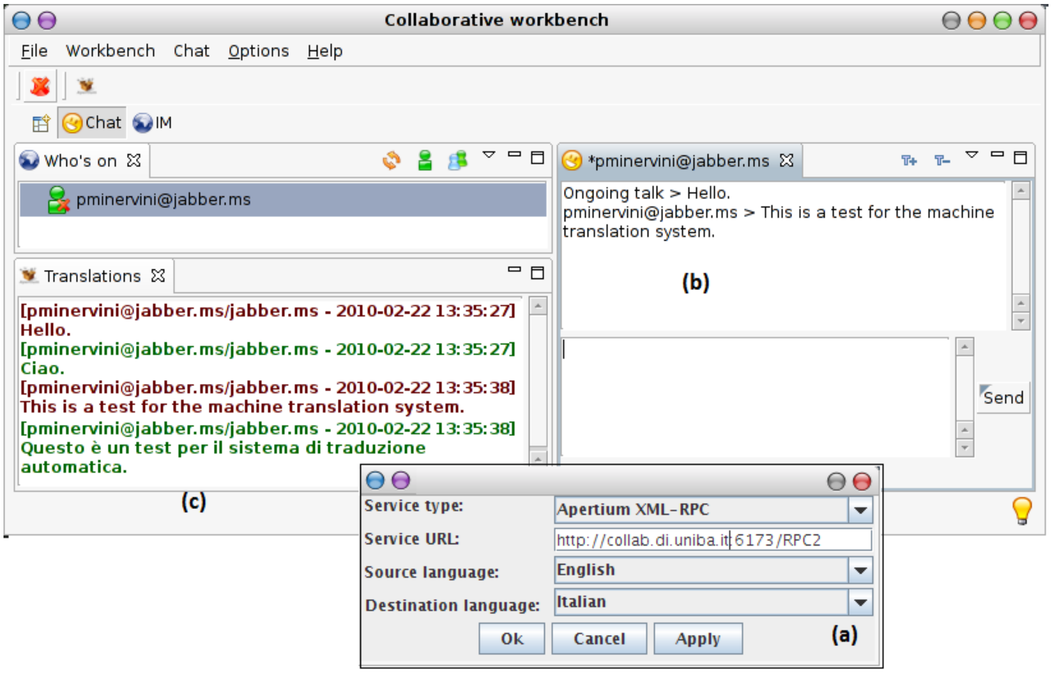


Figure 1. A screenshot of eConference showing instant messages automatically translated from English to Italian.

Accurate computer translation is particularly appealing because it is quicker, more convenient, and less expensive than human translators are. Military coalitions are another example of global teams suffering from multi-cultural and multi-language bottlenecks. Odgen [28] and Jones & Parton [11] provide two examples of employing instant messaging tools augmented with automatic machine translation, for helping military coalition partners to communicate using their own language. Recently, the EU commission funded the MOLTO project (Multi-lingual Online Translation)[3] with the goal of producing accurate machine-translations of the official documents and save the billion euro currently spent per year to translate them in the 23 official languages of the Union.

Finally, aside from research prototypes or projects, also commercial tools that offer cross-language chat services are available, such as IBM Lotus Translation Services for Sametime [4] and, lately, VoxOx [5], which provides cross-language translations for most of the existing instant messaging networks.

## IV. METHOD

The goal of the simulation was to evaluate the feasibility of adopting a MT service in a cross-language, real time, text-based chat. In particular, the simulation compared the performance (i.e. effectiveness and efficiency) of the two MT services described earlier, Apertium and Google Translate.

[3] http://www.molto-project.eu

[4] http://www-01.ibm.com/software/lotus/sametime

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

While the effectiveness of a MT service relates to the *quality* of the translated output (i.e. accuracy), the efficiency relates to the amount of *time* necessary to translate the original input text (i.e. speed). Efficiency is fundamental in our scenario because if the use of MT involves a large amount of additional time, then it would break the real-time feature of a chat and hamper the synchronous communication.

### A. *Evaluation of Translation Quality*

Evaluating the quality of a translation is an extremely subjective task and disagreements about evaluation methodology are rampant [1][23]. Nevertheless, evaluation is essential. In this study, we entailed four human raters to evaluate accurately each translation in terms of *intelligibility*, which is affected by grammatical errors, mistranslations and untranslated words [27].

In our simulation, the raters assessed the intelligibility of translations assigning scores to output sentences produced by the two MT services. The scoring scheme adopted is a 4-point Likert scale (see Table I), anchored with values *4 = completely unintelligible* and *1 = completely intelligible*. The scale, proposed in [16], seemed appropriate to our goal because it is not too fine grained (i.e. does not consist of too many values), it can be easily applied as descriptions are well defined (i.e. can be uniformly interpreted by evaluators), and there is no middle value (i.e. helps to avoid central tendency bias in ratings by forcing raters to judge the output as either intelligible or not) [10][18].

Before the official scoring session was held, the raters participated in a training session in which they become acquainted with the scale. The raters were all master students

TABLE I. INTELLIGIBILITY SCALE [16].

| Value | Description |
|---|---|
| 1 | ***Completely intelligible*** The sentence is perfectly clear and intelligible. It is grammatical and reads like ordinary text. |
| 2 | ***Fairly intelligible*** The sentence is generally clear and intelligible. Despite some inaccuracies or infelicities of the sentence, one can understand (almost) immediately what it means. |
| 3 | ***Poorly intelligible*** The general idea of the sentence is intelligible only after considerable study. The sentence contains grammatical errors and/or poor word choices. |
| 4 | ***Completely unintelligible*** The sentence is unintelligible. Studying the meaning of the sentence is hopeless; even allowing for context, one feels that guessing would be too unreliable. |

completing their thesis project in our laboratory, at the University of Bari, and were selected among those who proved to have a good knowledge of English.

### B. Simulation

The text corpus used to run the simulation is composed of chat logs, written in English and collected from five requirements workshops run during an experiment on the effects of text-based communication in distributed requirements engineering [5]. We used one workshop log (CL1) to train the raters, whereas the remaining four (CL2-CL5) were employed as the test set during the simulation. Overall, the test set accounted for over 2.000 utterances to be translated by both MT services. Participants in each workshop ranged from five to eight undergraduate students attending a requirements engineering course at the University of Victoria, Canada. During a workshop the participants, either acting as a client or as a developer, had first to elicit the requirements specification of a web application (first session); then, they had to negotiate and reach closure on the previously collected requirements (second session). Table II contains an excerpt of the chat logs, showing the messages exchanged between two clients and two developers.

As a first step, we modified our eConference MT plugin in order to process XML files containing the chat log entries. The plugin spawned several threads, one for each participant in the workshop, which processed the file and sent in chat messages. Each thread also received any message sent and then invoked the translation service one by one. Because all the messages in the logs are timestamped, we were able to send them with the same timing as in the real workshops, that is, we recreated a realistic condition similar to the one that would have happened if the real requirements workshops had relied on MT. Besides, we also put each translation service under the same stress condition in which messages sent at the same time would have caused the translation service to be invoked concurrently by each participant in the workshop.

The simulation was executed on a box running Debian Linux, with a 2GHz Dual-Core AMD Opteron CPU, and 4 GB of memory. Finally, in order to compare the performance of Apertium and Google Translate, the simulation was run twice on the same text corpus and on the same machine, once for each MT service.

## V. RESULTS

Our analysis focused on evaluating both the effectiveness and the performance in order to evaluate, respectively, the goodness of translations in terms of intelligibility, and the extra amount of time taken to translate the sentences from the original language to the target language.

### A. Translation Quality Results

The four coders performed the rating separately. We measured the inter-rater agreement by computing the Fleiss' Kappa index for multiple raters [9]. In particular, for the Apertium service, the Kappa index shows a fair agreement level (k=.37) [1]. Instead, .for Google Translate, the Kappa index measured shows a moderate agreement level between the raters (k=.47) [1].

In order to identify differences in the quality of translation produced by the two MT services, as perceived by the raters, we first evaluated how many sentences were evaluated as intelligible (i.e. belonging to categories 1 and 2) and unintelligible (i.e. belonging to categories 3 and 4). Figure 2 shows that, for Google Translate, over a half of the whole test suite (2053 sentences) was judged intelligible (63.3%). Conversely, for Apertium over the 62.2% of the translated sentences was judged unintelligible. In addition, we found that the mean and median ratings for Google Translate were, respectively, 2.17 and 2.0. Instead, for Apertium the mean and median ratings were 2.8 and 3.5, respectively.

Afterwards, we performed the paired t-test for two related samples. We summed the ratings from each rater for each translated utterances, thus obtaining N=2053 summed scores for each MT service. The summed scores obtained ranged between 4 (best) and 16 (worst). The paired t-test result, shown in Table III, revealed a statistically significant difference (p=.00) in favor of Google Translate, which thus was judged to produce more accurate translations than Apertium.

TABLE II. AN EXCERPT FROM THE CHAT LOGS.

| Student | Message |
|---|---|
| Client 1 | we don't necessarily need the conversations to be stored in a DB... |
| Client 2 | We also need application sharing. IE - letting someone else access a single window on my computer. |
| Client 1 | and yeah, we do need application sharing |
| Dev 1 | Ok |
| Dev 1 | we have questions about that so just wanted an overview |

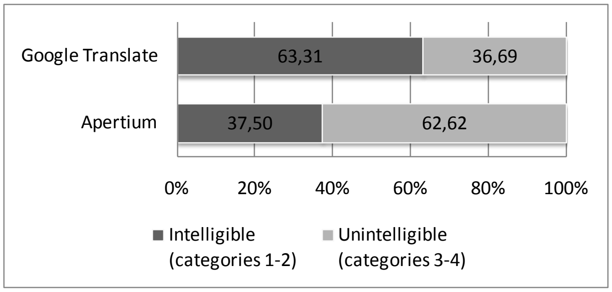


Figure 2. Percentage of intelligible vs. unitelligible ratings.

## B. *Time performance results.*

Each data point in the following two figures was obtained as an average measure of 256 repeated translation requests.

Figure 3 shows that Apertium response times are lower than those of Google Translate are. In fact, in the worst case, Apertium took less than 30 ms to complete, on the average, the repeated translation requests, whereas Google service took twice the amount of time (over 70 ms). Conversely, the graph of response times shows that Google Translate performance does not depend on the length of the sentences, as in the case of Apertium.

Figure 4 plots the response times of the two services when completing concurrent translation requests from an increasing number (from 1 to 8) of clients. The data points were again collected as an average measure of 256 repeated requests for translating the longest sentences available in the whole data set (362 characters). The graph shows that Apertium performances are better when the numbers of concurrent requests are low (less than 4), whereas Google Translate is better able to cope with a growing number of concurrent clients.

## VI. DISCUSSION

The two main results of our work are the assessment of the translation quality (i.e. accuracy) and the performance (i.e. speed) of the Google Translate and Apertium services.

With respect to the performance test, we found good time responses for both services, which also proved to scale up well as the number of clients – i.e. concurrent requests – and the length of sentences increase (see Figure 3 and 4, respectively). However, the time performance of Apertium (less than 30 ms in the worst case) is better than Google (around 70s ms in the worst case). This is because, Google service is publicly available (i.e. other people might be using it at the same time of our tests) and thus the load of requests to the well-known Google Translate service was reasonably much higher than that served by Apertium, which run instead as a private service. Nevertheless, we noticed that Apertium response time increases with the length of sentence, while Google Translate performance tends to be rather stable, independently of sentence length and concurrent requests.

TABLE III. RESULTS FROM THE PAIRED T-TEST.

| | **Mean** | **Std. Dv.** | **N** | **Diff.** | **Std. Dv. Diff.** | **t** | **df** | **p** |
|---|---|---|---|---|---|---|---|---|
| Apertium | 11.19 | 4.06 | 2053 | -2.58 | 4.23 | -27.06 | 2052 | 0.00 |
| Google Translate | 8.66 | 4.13 | | | | | | |

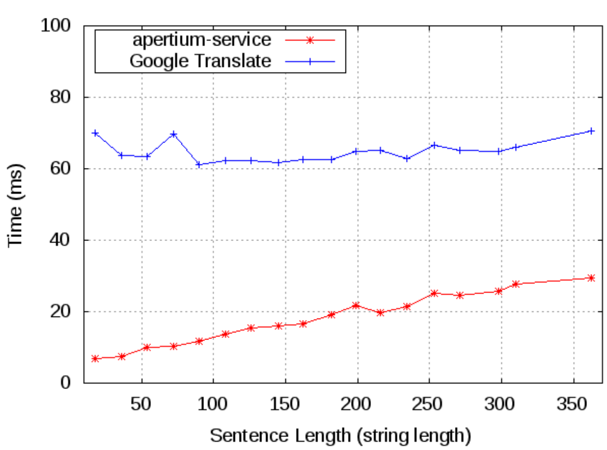


Figure 3. A comparison between the amount of time (in ms) taken by Google Translate and Apertium services to translate sentences of growing lengths.

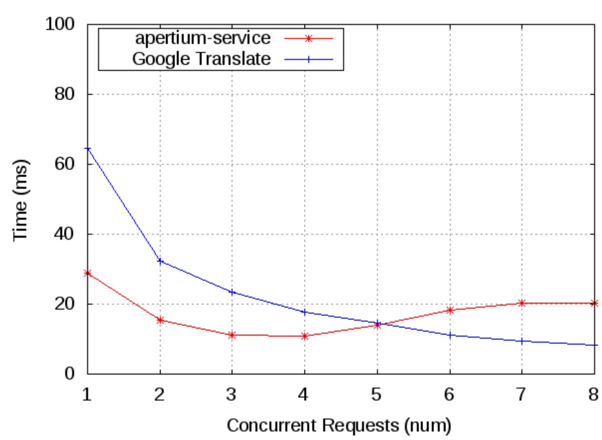


Figure 4. A comparison between the amount of time (in ms) taken by Google Translate and Apertium services to complete concurrent requests from an increasing number of clients.

With respect to the translation quality, we employed four raters to judge the intelligibility of the output produced by the two services. We then evaluated the inter-rater agreement by computing multiple Kappa index, which was measured 0.36 for Apertium and 0.46 for Google Translate. The fair to moderate Kappa values measured can be partially explained by having employed non-bilingual raters for the evaluation of translations quality. In fact, we employed four Italian master students who are not native English speakers, although knowledgeable in software engineering and thus fully able to understand the context of conversations. Hence, possible disparities in raters' language skill can probably account for the moderate agreement levels achieved.

Besides, Google Translate was found to produce significantly more accurate (i.e. more intelligible) translations than Apertium. On the average, a Google Translate translation is rated 2.17 (median 2.0), with over the 63% of translations falling into category 1 or 2, that is, judged to be fully intelligible by the raters. Conversely, Apertium average translation quality is rated 2.8 (median 3.5), with most of the translation produced (~63%) falling in category 3 or 4, that is, judged to be partially or completely unintelligible by the raters.

We identified a couple of reasons why Apertium achieved lower intelligibility ratings than Google. The first reason is the low quality of the translation rules defined in the English-Italian pair. We tried to address this issue using one of the five chat logs available – the one not used by raters for the evaluation – to add linguistic knowledge to the EN-IT pair[6] in Apertium by adding missing rules and words. Nevertheless, intelligibility ratings were not only lower than those produced by Google Translate, but also worse than those obtained using Apertium with the full-fledged English-Spanish pair, which we evaluated informally. The second one is that Google service was better able to cope with short, slang forms, typical of text-based chats, such as "*hes*" instead of "*he's*", "*dont*" instead of "*don't*", which instead, Apertium proved not to manage well.

Despite achieving better intelligibility results, Google Translates suffers from at least a couple of drawbacks. The first one is due to the statistical approach used, which prevents Google Translate from being improved, as in the case of rule-based systems, to which specific domain knowledge can be added in form of new dictionaries and translation rules. The second drawback is a limitation that raises privacy concerns. When using Google Translate, one cannot install the MT service on a company's private server, meaning that private data must be sent to Google servers for being translated.

Overall, these results suggest that machine translation services can be helpfully employed in multicultural context to reduce language disparity issues in a quick and convenient way. Obviously, the generalizability of the results from our study is limited by being a simulation. We identified at least three major threats.

First, our simulation involved only one-way translations, that is, utterances were only translated from English to Italian and not vice versa. Instead, a more realistic experiment would involve two-way translations between cross-language groups. One could reasonably argue that two-way translation would increase intelligibility issues. One way to overcome this limitation, is studying how machine translation affect the establishment of common ground, that is the mutual knowledge that people involved in a discussion share and the awareness of it. In fact, mutual knowledge and its awareness are both affected by the asymmetry of machine translation, which prevents the sender of a message to know whether it has been translated well and, consequently, accepted by the receiver [25].

Second, we evaluated translation quality exclusively in terms of intelligibility, that is, the raters evaluated the comprehensibility of a sentence in the context provided by the history of all previous messages. When evaluating translation quality other dimensions are often taken into account, such as style (or fluency), and accuracy (or fidelity). However, as Hutchins and Somers noted [16], style matters only when a translation is intelligible; furthermore accuracy scores are often closely related to the intelligibility scores since high intelligibility normally means high accuracy. On the contrary, it is more efficient to analyze just those cases where the output is rated incomprehensible, leading one to suppose something has gone wrong.

Finally, our study worked on the sentence as the unit of analysis and, consequently, the raters judged intelligibility of sentences according to the context, whereas in group-collaboration task performance is paramount. Hence, even though results from our simulation are somewhat encouraging, we can by no means hypothesize whether the translation quality of either MT service would be good enough to allow participants to complete a group task – in our scenario, allow stakeholders to define and negotiate software requirements for a small web application. Previous works in the field of MT (e.g. [27]) show that although employing machine translation does not prevent task completion, it considerably slows it down. Nevertheless, these works compared the interaction performance of machine-translated groups to those of standard groups on puzzle-like tasks execution. One can reasonably argue that greater concerns would arise during the execution of requirements engineering group tasks. Requirements engineering activities, such as elicitation and negotiation workshops, are complex, communication-intensive tasks that require specialized knowledge and techniques to be applied. As such, during their execution low quality translations could worsen or even cause misunderstandings, which in turns might generate defects in the requirements specifications.

## VII. Conclusions & Future Work

Global software projects are affected by the combination of geographical, temporal, and cultural distance [7]. While there is a growing literature about the effects of distance and time differences, we know little about how to handle intercultural factors [20]. In fact, work that takes place over long distances means that communication will often involve

[6] SVN revision 19832

distant cultures, with different languages and communication styles exacerbating communication problems [12].

To date research efforts have mostly focused on the organizational (i.e. processes and coordination) aspects of globally distributed labor [13], as well as on computer-mediated communication [8] and tools [32], but little on culture *per se*. In fact, only in the last decade research started to investigate on the specific issues of cultural difference in globally distributed projects [7][21][29].

In this paper we explored the idea of applying automatic, cross-language translation to communication-intensive activities, such as distributed requirements engineering, we compared two successful MT services, which entail two completely opposite approaches, namely rule-based (Apertium) and corpus-based (Google Translate).

In our simulation, we used chat logs collect from five distributed requirements engineering sessions. The logs were first translated from English to Italian and then translation quality was evaluated by multiple human raters in terms of intelligibility. Our findings show Google Translate produces significantly more intelligible translations than Apertium. Besides, we also tested the performance in terms of the amount of time requested to translate sentences with multiple concurrent requests to the MT services. The rather small amount of extra time necessary to translate concurrently chat messages (about 70 ms on in the worst case observed) shows that state-of-the-art MT services can be embedded into synchronous text-based chat without disrupting real-time interaction.

As future work we intend to set up a controlled experiment rather than a simulation, so that: (1) both cross-language groups and same language groups of participants can be compared while interacting to complete a knowledge- and communication-intensive task, such as a requirements elicitation or negotiation workshop; (2) a MT service is used for two way-translations; (3) language pairs other than English-Italian are used, which are more relevant to the global software development scenario (e.g. English-Portuguese).

## ACKNOWLEDGMENT

We would like to thank the students who performed the evaluations of the translation quality.

## REFERENCES

[1] D. G. Altman, *Practical Statistics for Medical Research*, Chapman and Hall, London, 1991.

[2] C. Armentano-Oller, A. M. Corbì-Bellot, M. L. Forcada, M. Ginestì-Rosell, B. Bonev, S. Ortiz-Rojas, J. A, Perez-Ortiz, G. Ramirez-Sanchez, and F. Sanchez-Martinez, “An open-source shallow-transfer machine translation toolbox: consequences of its release and availability,” Proc. workhsop on Open-Source Machine Translation (OSMaTran), Machine Translation Summit X, Phuket, Thailand, pp. 23–30, 2005.

[3] D. Arnold, “Why translation is difficult for computers”, In *Computers and Translation: A translator's guide*. Benjamins Translation Library, 2003.

[4] Bangalore, S., Murdock, V., and Riccardi, G. “Bootstrapping bilingual data using consensus translation for a multilingual instant messaging system.” *Proc. 19th Int’l Conference on Computational Linguistics (COLING)*, Taipei, Taiwan, Aug. 24 – Sep. 1 2002, Volume 1, doi:10.3115/1072228.1072362.

[5] F. Calefato, D. Damian, and F. Lanubile, "An Empirical Investigation on Text-Based Communication in Distributed Requirements Engineering", *Proc. 2nd Int’l Conf. Global Software Engineering (ICGSE ’07)*, Munich, Germany, 27-30 August, 2007, doi: 10.1109/ICGSE.2007.9.

[6] F. Calefato and F. Lanubile, "Using Frameworks to Develop a Distributed Conferencing System: An Experience Report", *Software: Practice and Experience*, 2009, vol. 39, no. 15, pp. 1293–1311, doi: 10.1002/spe.937.

[7] E. Carmel, and R. Agarwal, “Tactical Approaches for Alleviating Distance in Global Software Development,” *IEEE Softw.*, vol. 18, no. 2, pp. 22-29, Mar. 2001, doi:10.1109/52.914734.

[8] D. Damian, F. Lanubile, and T. Mallardo, "On the Need for Mixed Media in Distributed Requirements Negotiations", *IEEE Transactions on Software Engineering*, Vol. 34, No. 1, January 2008, pp. 116-132.

[9] J. L. Fleiss. *Statistical methods for rates and proportions*. 2nd ed. New York: John Wiley, 1981, pp. 38–46

[10] R. Garland. “The Mid-Point on a Rating Scale: Is it Desirable?”, *Marketing Bulletin*, Vol. 2, 1991, pp. 66-70.

[11] S. Jones and G. Parton. “Collaboration Across the Multinational Battlespace in Support of High-stakes Decision Making - Instant Messaging with Automated Language Translation”, Technical report, The Mitre Corporation, 2008.

[12] J.D. Herbsleb, and D. Moitra, “Guest Editors' Introduction: Global Software Development,” *IEEE Softw.*, vol. 18, no. 2, 2001, pp. 16-20.

[13] J.D. Herbsleb. “Global Software Engineering: The Future of Socio-technical Coordination,” *Future of Software Engineering (FOSE’07)*, Washington, DC, May 23 - 25, 2007, pp. 188-198, doi:10.1109/FOSE.2007.11.

[14] C. Hogan and R. Frederking, “WebDIPLOMAT: a Web-based interactive machine translation system.” *Proc. 18th Int’l Conference on Computational Linguistics - Volume 2*, Saarbrücken, Germany, Jul. 31 – Aug. 04, 2000, pp. 1041-1045, doi:10.3115/992730.992801.

[15] H. Holmstrom, E. O. Conchuir, P. J. Agerfalk, and B. Fitzgerald, "Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance," 1st Int’l Conf. on Global Software Engineering (ICGSE '06), Florianopolis, Brasil, Oct. 2006, pp.3-11, doi:10.1109/ICGSE.2006.261210.

[16] W. J. Hutchins and H. L. Somers, *An Introduction to Machine Translation*, Academic Press, 1992.

[17] Y. Hsieh, “Culture and Shared Understanding in Distributed Requirements Engineering,” 1st Int’l Conf. on Global Software Engineering (ICGSE’06), Florianopolis, Brazil, Oct. 2006.

[18] R. Johns. “One Size Doesn't Fit All: Selecting Response Scales For Attitude Items.” *Journal of Elections, Public Opinion, and Parties*, Vol. 15, No. 2, 2005, pp. 237-264.

[19] D. Jurafsky and J. H. Martin, “Speech and Language Processing 2nd ed.,” Prentice Hall Series in Artificial Intelligence, Prentice Hall, 2008.

[20] P. Kruchten, “Analyzing intercultural factors affecting global software development - a position paper,” 3rd Int’l Workshop on Global Software Development (GSD 2004), Edinburgh, Scotland, UK, 24 May 2004, doi:10.1049/ic:20040315.

[21] A.E. Milewski, M. Tremaine, F. Köbler, R. Egan, S. Zhang, and P. O'Sullivan, “Guidelines for effective eridging in global software engineering,” *Software Process: Improvement and Practice*, vol. 13, no. 6, 2008, pp. 477-492.

[22] P. Minervini, “Apertium goes SOA: an efficient and scalable service based on the Apertium rule-based machine translation platform,” Proc. 1st Int’l Workshop on Free/Open-Source Rule-Based Machine Translation, Alacant, Spain, Nov. 2-3, 2009, pp. 59–66.

[23] R. Mitkov, “The Oxford Handbook of Computational Linguistics,” Oxford Handbooks in Linguistics S., Oxford University Press, 2003.

[24] N. Yamashita and T. Ishida. “Effects of machine translation on collaborative work.” *Proc. 20th Int’l Conference on Computer Supported Cooperative Work* (CSCW '06), Banff, Alberta, Canada, November 04-08, 2006, pp. 515-524, doi:10.1145/1180875.1180955.

[25] N. Yamashita, R. Inaba, H. Kuzuoka, and T. Ishida. “Difficulties in establishing common ground in multiparty groups using machine translation.” *Proc. 27th Int’l Conf. on Human Factors in Computing Systems (CHI '09)*. Boston, USA, April 4-9, 2009, pp, 679-688, doi:10.1145/1518701.1518807.

[26] F. J. Och and H. Ney, The alignment template approach to statistical machine translation. *Computational Linguistics*, vol. 30, no. 4, pp. 417-449, 2004.

[27] W. Ogden, R. Zacharski, S. An and Y. Ishikawa, “User choice as an evaluation metric for web translation in cross language instant messaging applications,” Proc. Machine Translation Summit VII, Ottawa, Canada, Aug. 2009.

[28] W. Odgen. “A Task - Based Evaluation Method for Embedded Machine Translation in Instant Messaging Systems,” in *Advanced Decision Architectures For The Warfighter: Foundations and Technology* (P. Mcdermott And L. Allender eds.), chapter 19, pp. 341-357, Aug. 2009.

[29] J. S. Olson and G. M. Olson, Culture Surprises in Remote Software Development Teams, *ACM Queue*, vol. 1, no. 9, Dec. 2003, pp. 52-59, doi:10.1145/966789.966804.

[30] L.D. Paulson, “Translation technology tries to hurdle the language barrier,” *Computer*, vol. 34, no. 9, 2001, pp. 12-15.

[31] F. Sánchez-Martínez, and M. L. Forcada, “Inferring shallow-transfer machine translation rules from small parallel corpora,” *Journal of Artificial Intelligence Research,* vol. 34, p. 605-635.

[32] H. Spanjers, M. ter Huurne, B. Graaf, M. Lormans, D. Bendas, R. van Solingen, "Tool Support for Distributed Software Engineering," *Int’l Conf. Global Software Engineering. (ICGSE '06)*, Oct. 2006, pp.187-198.