How Does Machine Translation of User Interface Affect User Experience? A Study on Android Apps

Xue Qin[1], Smitha Holla[1], Xiaoyin Wang[1], Liang Huang[2]
[1] Department of Computer Science, University of Texas, San Antonio, TX
[2] Department of Computer Science, Oregon State University, Corvallis, OR

September 22, 2015

Abstract

For global-market-oriented software applications, their user interfaces need to be translated to local languages to facilitate users from different areas in the world. A long-term practice in software industry is to hire professional translators or translation companies to perform the translation. However, due to the large number of user-interface labels and target languages, this is often too expensive for software providers, especially cost-sensitive providers such as personal developers of Android apps. On the other hand, more and more mature machine translation techniques are providing a cheap though imperfect alternative, and the Google Translation service has been widely used for translating websites and apps. However, the effect of translation quality of GUI labels on user experience has not been well studied yet. In this paper, we present a user study on 6 popular Android apps, to have 24 participants perform tasks on app variants with 4 different translation quality levels and 2 target languages: Spanish and Chinese. From our study, we acquire the following 3 major findings, including (1) although sharing only about 30% of GUI labels with the original localized versions, machine translated versions of GUI have similar user experience on most studied aspects, except a slightly lower task completion rate; (2) manually enhanced but still imperfect machine translated versions are able to achieve exact the same user experience in almost all studied aspects; and (3) users are not satisfied with the GUI of machine translated versions and the two major complaints are misleading labels of input boxes, and unclear translation of items in option lists.

1 Introduction

In this era of globalization, most software applications have potential users from different regions of the world. Emerging techniques, such as the Internet and smartphone app markets, make it easier than ever to distribute software applications globally. To better serve users from diverse cultural and linguistic backgrounds, software providers

need to prepare versions in different locales, and the user interface of these versions should be in local language.

While manual translation has long been the practice in software industry, the recent progress in machine translation provides an imperfect, but cheaper alternative, which helps small software companies and personal software developers to generate localized versions of user interface, as well as helps software users to use software without local versions. As an example, Google Translation Service[1] has long been used for translating websites and smartphone apps.

However, the effect of machine translated user interface on software user experience remain largely unknown. It is well known that machine translation results are of lower readability, but usability of user interface can be affected by more factors. First of all, GUI labels with misleading meanings may result in accumulation of erroneous operations and amplify the translation errors. Second, users may be able to easily guess the meaning of UI labels through its context in the same interface. Third, a lot of pictures and symbols are also used in user interface, and they can provide extra hints for software users. For example, a symbol "+" is often used on a button to add elements such as adding contacts or new account, and the symbols are typically the same in all locales because their meaning can be understood all over the world.

In this paper, to better understand how lower-quality translation of GUI affects user experience, we present a user study on Android apps from different domains and with different translation quality. In particular, we used 6 popular Android apps as subjects and considered 4 different versions for each of them: the manually translated version (*Original*), the Google-translation based version (*Google*), the combined version with half labels translated with Google-translation service and the other half of labels translated manually (*Half*), and the French version as the not-translated version (*French*). We consider Spanish and Chinese as our target language for translation. We performed the study with 12 Spanish native speakers and 12 Chinese native speakers, and each participant worked on 4 different translation-quality-level versions of 4 different apps (1 version of each app).

This paper mainly makes the following three contributions:

- A study on how GUI translation quality affects user experience.
- An automatic tool for generating Android apps with different translation quality levels.
- A data set including different versions of Android apps, tasks on the apps, recorded videos and surveys, which serves as the foundation for future research on this topic

2 Study Design

In our study, we try to answer the following three research questions. Through answering these questions, we are able to better understand the effect of low translation

[1] https://translate.google.com/

Table 1: Subject Android Apps

App Name	Domain	Downloads
k9	Email Client	5 - 10M
1Weather	Weather Report	10 - 50M
BrainTrainer	Game	1 - 5M
MyFitness	Health	10 - 50M
Line	Social and Communication	100 - 500M
ResumeMaker	Jobs	10 - 50K

quality on both objective and subjective user experience.

- **RQ1:** Are users able to correctly finish tasks on apps with lower GUI-translation quality?
- **RQ2:** How efficiently are users when they perform tasks on apps with lower GUI-translation quality?
- **RQ3:** What is the satisfaction of users on apps with lower GUI-translation quality?

Selection of Participants. In our study, we recruited 24 participants, and all of them are students from different departments in a University. Such a selection criterion may limit our findings to students, but students are an important group of smartphone users who will be in different occupations in the future. Furthermore, the background and ability of University students are relatively similar, thus limit our participant to students helps remove noises brought in by ability / background difference of participants. To judge the quality of the user interface, we require the participants to be Spanish or Chinese native speakers.

Selection of Android Apps. In our study, we selected 6 Android Apps, and their information is in Table 1. These apps are from different domains, and are all near-top apps in their domains. Actually, to avoid our study being affected by user previous experience with subject apps, we choose only participants who have never used the subject apps before they perform the study. Therefore we can not choose the top apps because they have been used by so many users, and we chose near-top apps so that they are still popular enough to be representative. Since we need to decompile and recompile the apks files to generate versions with different translation qualities, we used only apps whose apk files can be decompiled or recompiled without errors.

Different Translation Quality. In our study, we considered four translation quality levels: manual translation, half manual translation, machine translation, and no translation. We studied half manual translation because we want to generate an intermediate quality level between the current machine translation and manual translation, and want to check how a more advanced but still imperfect translation technique will affect user experience. It should be noted that, we did not use the English version as the untranslated version, because it is very difficult to find participants who do not know English at all in United States. Therefore, we used the French version as the untranslated version, to simulate the scenario where the users do not understand string labels at all.

To prepare app versions with machine translation and half machine translation quality level, we decompile the apk files and extracted the resource file that stores GUI la-

Figure 1: Task Completion Rate for Spanish Versions

Table 2: Designed Tasks

App Name	Task
k9	compose, reply, forward and spam
1Weather	read current weather and warning
BrainTrainer	complete a game
MyFitness	create food with recipe and add to meal
Line	invite a friend and modify settings
ResumeMaker	make a resume

bels (i.e., "strings.xml"), and replaced the values in the file with corresponding translation results (e.g., Google, Half). Our evaluation further shows that, the Google version shares about 30 % of UI label strings with manual translation, and the Half version shares about 60 %.

Tasks for Users. For each subject app, we designed several tasks for the participants to work on. When designing the task, we tried to simulate the basic usages of the app. The designed tasks are presented in Table 2. Note that due to feature difference, different numbers of tasks are designed for each app. Also, for ResumeMaker, to protect privacy and maintain task consistency, we provide information for participant to put into their resumes.

User Survey. To further acquire the users' feelings and satisfaction of different versions of apps, we further designed a survey for the users to answer after they performed the tasks. It should be noted that, the users did not know what the translation-quality level of the apps they played with. The questions are listed as follows. Specifically, Q1 to Q3 are questions about participants' experience on the GUI. Q4, and Q5 are about

Table 3: Assignments of Tasks to Participants

App/Version	Level 0	Level 1	Level 2	Level 3
k9Mail	P1	P6	P5	P4
One Weather	P2	P1	P6	P5
Brain Trainer	P3	P2	P1	P6
MyFitnessPal	P4	P3	P2	P1
Line	P5	P4	P3	P2
ResumeMaker	P6	P5	P4	P3

overall feeling of the app, and Q6 asked the participant to provide detailed information.

- Q1. How well were you able to understand the user interface of the app?
- Q2. Do you have any misunderstanding (that you realized later) in the user interface?
- Q3. Were you able to find the information you were looking for? Please specify if there are any app function such as a button or a menu that you feel hard to find.
- Q4. Will you use the app (in the current shape) in future?
- Q5. What is your overall satisfaction of the app's user interface?
- Q6. Detail the difficulties when performing the given task.

Study Process. In our study, one participant should not work on different versions of a same app. Otherwise, her experience on the app will affect her performance in the following tasks. Therefore, we have each user working on four different versions of four different apps. Specifically, we put 6 participants in a group, and for each group, we assign tasks to participants as illustrated in Table 3.

3 Study Results

In this section, we present our study results, and organize them by which research question is answered.

Task Completion. To answer **RQ1**, for each app, we check whether all the tasks are correctly finished. The results are presented in Figure 1 and Figure 2. Actually, in all except French versions, participants are able to finish all tasks, but they may make mistakes such as filling address into the input box of occupation, or changing the wrong settings. In French versions, many participants simply could not finish a large portion of the tasks.

From the figures, we can see that, for both languages, all 3 translated versions have very close task completion rate, although that for Google version is slightly lower than the other two. In BrainTrainer, all 4 versions achieve 100% task completion, because BrainTrainer is a game with a lot of pictures which makes even its French version usable for participants.

Usage Efficiency. To answer **RQ2**, for each task, we record the time of completing each task. The results are presented in Figure 3 and Figure 4. Since the time to finish

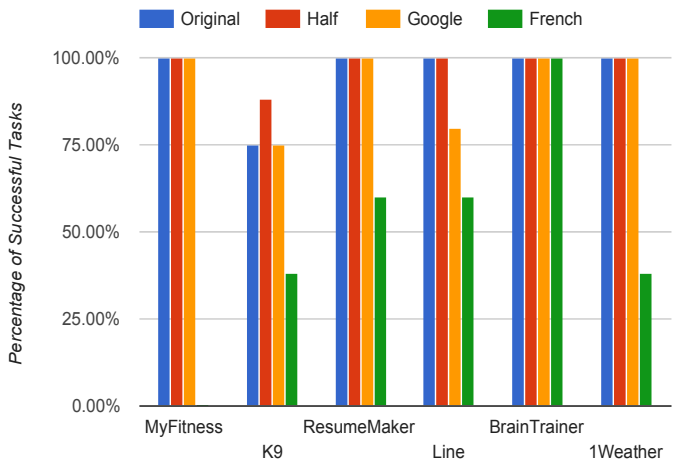

Figure 2: Task Completion Rate for Chinese Versions

tasks for different apps varies a lot, we use the completion time of the original version as the standard (i.e., 1), and calculate the relative time of other versions.

From the figures, we can see that, the original versions always have the shortest task completion time, and the French versions always have the longest. The comparison between the Google version and the Half versions is opposite in Spanish and Chinese. Actually, our later calculation shows no significant different among all 3 translated versions. Although the French versions have task completion time to translated versions, the reason is actually participants gave up in most cases, so the completion time becomes short.

User Satisfaction To answer **RQ3**, for each task, we did statistics on our survey responses. The results are presented in Figure 5 and Figure 6. Since our questions Q1 to Q4 are all boolean questions, for the ease of presentation, if the answer is positive for user experience, we set the answer as 1, and otherwise 0. So we can calculate the average of answers as a score between 0 to 1. For Q5, we also map the rating 1-10 to the range of 0 to 1. We do not include the French versions, because the UI-related questions are not applicable, and their overall ratings are all very bad. From the figures, we can see that, the Google version is lower than the original version in Q1 for both languages, and in Q2 for Chinese. These results show that users have difficulties understanding the GUI and have been misled. Therefore, we can infer that, although users are able to guess the GUI labels quickly and finish the tasks, they are not comfortable with the GUI.

User Complaints To better understand why users are unsatisfactory with the machine translated GUI, we further investigated the response to Q6, and categorize the answers. In total, participants provided 22 complaints to Google and Half versions, in which the two major types are misleading labels (9 complaints), and finding options (6 complaints). The most complained labels are input box labels, because the meaning of

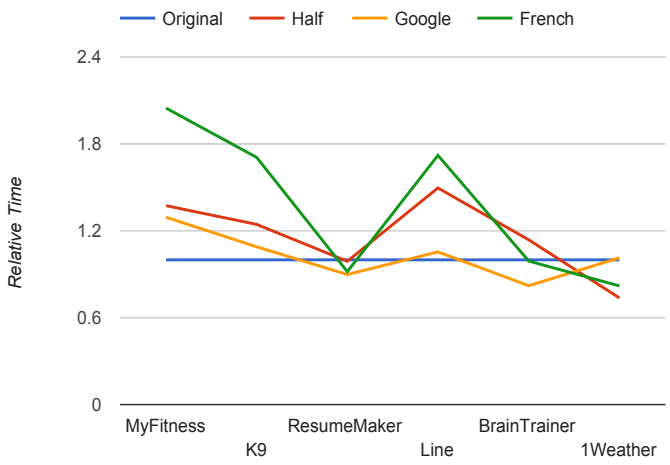

Figure 3: Task Completion Time for Spanish Versions

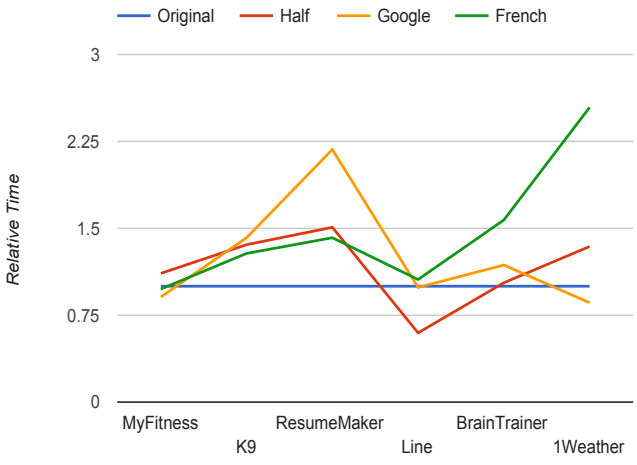

Figure 4: Task Completion Time for Chinese Versions

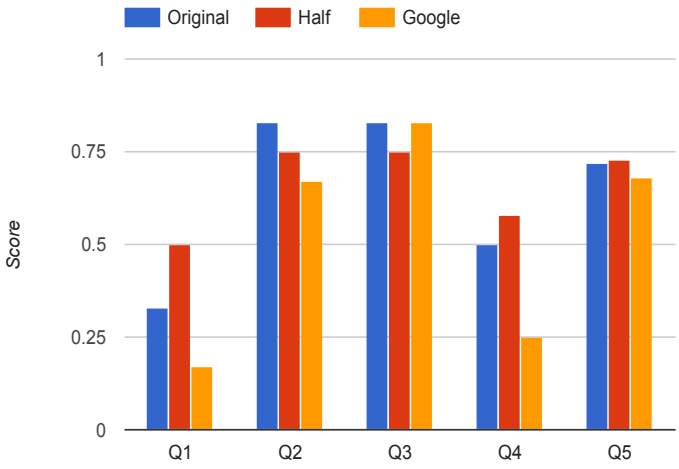

Figure 5: User Satisfaction of Spanish Versions

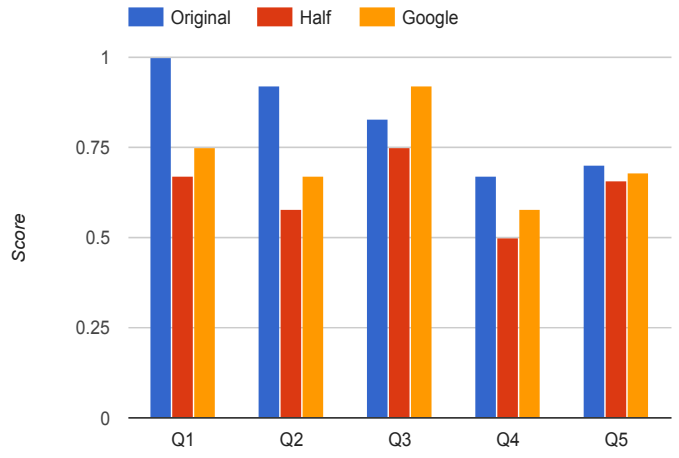

Figure 6: User Satisfaction of Chinese Versions

Table 4: Statistical Significance of Study Results

Study	O-H		H-G		G-F	
	ES	CH	ES	CH	ES	CH
Completion Rate	=	=	>	>	>	>
Completion Time	=	=	=	=	<	=
Q1	=	>	>	=	N/A	N/A
Q2	=	>	=	=	N/A	N/A
Q3	=	=	=	=	N/A	N/A
Q4	=	=	=	=	N/A	N/A
Q5	=	=	=	=	N/A	N/A

button labels are easier to guess through its behavior. Also, apps often have a long list of options, so mis-translation of items largely affects the searching of a specific item.

Statistical Significance. Finally, we apply statistical analysis to check which of our observations are of statistical significance. Specifically we compare various results of neighboring translation quality levels (O-H means Original compared with Half, H-G means Half compared with Google, and G-F means Google compared with French). We used T-test [1] for Task Completion Rate / Time and Q5, because they involve list of numbers. We used A/B Test [1] for Q1 to Q4 because they involve simply boolean values. We use 0.9 for confidence in both tests, and the results are presented in Table 4. From the table, we can see that, French versions are significantly worse than translated versions on most results. As we discussed above, half versions are significantly better than Google versions on task completion rate, and original versions are significantly better on Q1 and Q2.

Limitations. Our study has the following major limitations. First, various factors other than translation quality may affect our results. In particular, to avoid the effect of experience, we cannot have a same user to work on different versions of a same app. However, when different users are involved in the comparison, the users' ability and characteristic (e.g., quick finger response) may affect the results. To reduce such effect, we have each user work on all different versions to make sure the effect of their specialty is applied equally to all versions. However, it can still be the case that the user's specialty is app specific (e.g., good at games). Second, we chose students as our participants to generate a consistent user set, but it may also limit our study results to the student user group. Third, we used 6 Android apps as subjects and our study results may be specific to these Android apps. Therefore, we select them from different domains to enhance representativeness.

4 Related Works

As far as we know, this is the first user study on GUIs with different translation qualities. Several previous efforts reported that Google Translation is not sufficient for daily usage software [10] [9], but both works are based on human review of translation results instead of user studies on software GUI.

The concept of software localization [7, 2] appears when large software companies begin to seek market outside English-speaking countries. During the past twenty years, there have been some books [6, 15, 5] introducing some common guidelines on how to internationalize and localize a software application. Many programming reference books [13, 12, 11] also have one chapter or several sections briefly introducing the key ideas or approaches to performing internationalization and localization for certain programming languages.

On engineering of translatable interfaces, Tschernuth et al. [14] tried to unify translation of different platforms through a general and context-aware navigation tool set. Dixon et al. [3, 4] built a structured UI representation with pixel-based manipulation. This can be leveraged to automatic translation. Recently, Leiva and Alabau [8] proposed a novel framework for Just-in-time localization of web interfaces.

5 Conclusion

In this paper, we present a user study with 24 participants performing tasks on 6 Android apps. We generated 4 versions with different GUI-translation quality levels for each app, and each participant performed pre-defined tasks on 4 versions from 4 different apps. Our study found that, participants' task-completion time and correctness on machine-translated versions are close to those on manually translated versions. Participants' task-completion time and correctness on half-machine-translated versions have no significant difference with manually translated versions. However, machine-translated versions get lower score on UI satisfaction, and the major complaints are misleading labels of input boxes, and unclear translation of option items.

References

[1] 1994. Chapter 36 Large sample estimation and hypothesis testing. Handbook of Econometrics, Vol. 4. Elsevier, 2111 – 2245.

[2] V. Dagiene and R. Laucius. 2004. Internationalization of open source software: framework and some issues. In *2nd International Conference on Information Technology: Research and Education*. 204–207.

[3] Morgan Dixon and James Fogarty. 2010. Prefab: Implementing Advanced Behaviors Using Pixel-based Reverse Engineering of Interface Structure. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. 1525–1534.

[4] Morgan Dixon, Daniel Leventhal, and James Fogarty. 2011. Content and Hierarchy in Pixel-based Methods for Reverse Engineering Interface Structure. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. 969–978.

[5] Bert Esselink. 2000a. *A Practical Guide to Localization*. Benjamins, John Publishing Company.

[6] B. Esselink. 2000b. *A Practical Guide to Software Localization: For Translators, Engineers and Project Managers*. John Benjamins Publishing Co.

[7] Z. He, D. W. Bustard, and X. Liu. 2002. Software Internationalisation and Localisation: Practice and Evolution. In *Proceedings of the Inaugural Conference on the Principles and Practice of Programming and the Second Workshop on Intermediate Representation Engineering for Virtual Machines*. 89–94.

[8] Luis A. Leiva and Vicent Alabau. 2015. Automatic Internationalization for Just In Time Localization of Web-Based User Interfaces. *ACM Trans. Comput.-Hum. Interact.* 22, 3 (2015), 13:1–13:32.

[9] Manuel A. Pérez-Quiñones, Olga I. Padilla-Falto, and Kathleen McDevitt. 2005. Automatic Language Translation for User Interfaces. In *Proceedings of the 2005 Conference on Diversity in Computing*. 60–63.

[10] Ahmed Shwan and Muhammad Murtaza. 2012. Master's Thesis: Guidelines for Multilingual Software Development. (2012).

[11] Robert Simmons. 2009. *Hardcore Java*. O'Reilly Media.

[12] Nicholas A. Solter and Scott J. Kleper. 2005. *Professional C++*. Wrox.

[13] D. Ryan Stephens, Christopher Diggins, Jonathan Turkanis, and Jeff Cogswell. 2005. *C++ Cookbook*. O'Reilly Media.

[14] Michael Tschernuth, Michael Lettner, and Rene Mayrhofer. 2012. Unify Localization Using User Interface Description Languages and a Navigation Context-aware Translation Tool. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '12)*. 179–188.

[15] E. Uren, R. Howard, and T. Perinotti. 1993. *Software Internationalization and Localization, An Introduction*. Van Nostrand Reinhold.