Universiteit Utrecht

MBI Master Thesis

Using crowdsourcing for enterprise software localization

Introducing a method for quality- assurance and improvement

Abstract

In recent years platforms such as Mechanical Turk, Crowdflower and Innocentive have been a driving force behind the growing popularity of crowdsourcing. Crowdsourcing (or crowd computing) is becoming a common solution to provide answers to complex problems by automatically coordinating the potential of machines and human beings working together. However, research shows that several challenges still separate crowdsourcing from its generalized acceptance by industry. Most crucial factor being the quality delivered by the workers in the crowd and the difficulty of measuring it. This, coupled with the fact that current quality assurance techniques have been unsatisfactory, enterprises have been reluctant to fully adopt this new technology.

In this thesis, we discuss these issues, propose a new methodology for quality assurance and quality improvement and present a real industrial problem, through software localization (translation), in which the proposed solutions are put into practice. Results show that the proposed methodology significantly improves the quality of the translation, different configurations of users show successes in verifying this work and reaching professional quality translations are within sight.

Version: 1.0
Date: 25-3-2013
Author: Frederik Mijnhardt
Supervision: dr. Remko Helms
dr. Marco Spruit

Page intentionally left blank

Author: Frederik Mijnhardt
E-mail: a.f.mijnhardt@students.uu.nl
Student #: 3744248

Program: Business Informatics
Enrolled in: 2011
Period: April 2010 - March 2011

Subject: Crowdsourcing
Research organization: CA Technologies
Company supervision: dr. Victor Muntés-Mulero
Research supervision: dr. Remko Helms & dr. Marco Spruit

Version information

Version	Issued	Remarks	Author
0.1	21-3-2012	First Proposal	Frederik Mijnhardt
0.3	5-5-2012	Experiment Proposal	Frederik Mijnhardt
0.3	9-5-2012	Review by Remko	Remko Helms
0.7	23-12-2012	Preliminary Draft	Frederik Mijnhardt
0.7	27-2-2013	Review by Remko	Remko Helms
0.9	18-3-2013	Draft version	Frederik Mijnhardt
0.9	22-3-2013	Review by Remko	Remko Helms
1.0	25-3-2013	Final version	Frederik Mijnhardt

Glossary

Item	*Abbreviation*	*Description*
Machine Translation	MT	The process of using software to provide automatic translations from a source text to a target text
Source Text	T_{source}	The original text to be translated into a target language
Target Text	T_{target}	The translated text, translated from a source language
Corpus		A collection of texts that is used for research purposes
Natural Language Processing	NLP	Field of research concerned with the interaction between linguistics and computers
Computational Linguistics	CL	Field of research concerned with the interaction between linguistics and computers
Crowdsourcing		Crowdsourcing is an online, distributed problem-solving and production model
Crowd Computing		Method for achieving large-scale distributed computation through opportunistic networks
Localization		Translating and adapting a product for a specific culture and/or geographical location
BLEU		Automated machine translation evaluation technique based on n-grams technology
METEOR		Automated machine translation evaluation technique
Word Error Rate	WER	Metric for evaluating machine translations based on the levenshtein distance
Levenshtein distance		A text comparison mechanism for calculating the difference between two sequences
Linguistics		The field of languages and translation
Lexicon		A catalogue or collection of word for a specific language
Grammar		A set of rules that govern the composition of words and verbs
Syntax		The arrangement of sentences
Semantics		The meaning of a text or sentence and what they stand for
Translation Quality Index	TQI	A method to quantify the quality of a text
Total Error Points	TEP	The total errors in a text, weighted after the severity of the errors

Contents

Glossary **3**

1 Introduction **10**

1.1 Introduction 10

1.2 Relevance 12

1.3 Deliverables 12

1.4 Principal 13

1.5 Structure of Document 13

2 Research Method **14**

2.1 Introduction 14

2.2 Environment 15

2.3 Knowledge Base 15

2.4 IS Research 16

2.4.1 Design and Build 16

2.4.2 Evaluate 16

3 Theoretical Framework **18**

3.1 Introduction 18

3.2 Crowdsourcing 18

3.2.1 Crowdsourcing, Crowdcomputing and Human-based-computing 18

3.2.2 Crowdsourcing Platforms 21

3.2.3 Quality Assurance of crowdsourced work 22

3.2.4 Crowdworkers and their motivation 24

3.3 Linguistics 25

3.3.1 Linguistics 26

3.3.2 Translation 26

3.3.3 Machine Translation 27

3.3.4 Evaluation of machine translation and translation 28

3.4 Crowdsourcing and Linguistics 30

3.5 Gap in current Quality Assurance techniques 34

4 Crowdtranslation Platform **35**

4.1 Introduction 35

4.2 Translation Workflow 35

4.2.1 Current Translation Workflow 35

4.2.2 Proposed Crowdsource Translation Workflow 37
4.3 A Quality Assurance mechanism for translation tasks 38
4.3.1 AVI-Units for Quality Assurance 38
4.3.2 Post Edition and Reviewing combined with AVI-units 42
4.3.3 Worker Ranking for Quality Assurance 44
4.4 Rewarding Mechanism 45
4.4.1 Cost of translation 46
4.5 Prototype 47
5 Platform and AVI-unit evaluation 51
5.1 Introduction 51
5.2 Hypotheses 52
5.2.1 Hypothesis regarding the dataset and differences between languages 52
5.2.2 Hypotheses regarding the performance of the AVI-unit 53
5.2.3 Hypotheses measuring the performance of the verifiers in the AVI-unit 54
5.3 Experiment Design 55
5.3.1 Variables 55
5.3.2 Experiment Design 60
5.3.3 Selecting, Assigning and User Segmentation 62
6 Results 63
6.1 Introduction 63
6.2 Determining the crowdsourcers quality ranks 63
6.3 Experiment results 67
6.3.1 Crowdsourcer demographics 67
6.3.2 The impact of languages on translation quality 69
6.3.3 The AVI-unit decreases the number of errors 70
6.3.4 The quality output of the Post Edition AVI-unit 73
6.3.5 The performance of verifiers in the AVI-unit 78
6.3.6 The impact verifiers have on fixing errors in the AVI-unit 82
6.4 Results Summary 83
7 Discussion 85
8 Conclusion 86
9 Future research 88
10 Food for thought 89

11 System Recommendations 89

12 Acknowledgements 90

13 References 91

13.1 Appendix A: System Design: System Architecture 96

13.2 Appendix B:System Design: Entity Relation Diagram 97

13.3 Appendix C:Translation Quality Review Sheet 98

13.4 Appendix D: Translation Quality Error Categories and Types 99

13.5 Appendix E:Translation Quality Severity Levels 102

13.6 Appendix F: Platform Design: Error Mapping 104

13.7 Appendix G: Platform Design: Post Editor and Reviewer ranking 106

13.8 Appendix H: Platform Design: Verifier ranking 110

13.9 Appendix I: Platform Design: Calculating user rank over time 112

13.10 Appendix J: Experiment: Translation Capability Test 115

13.11 Appendix K: Experiment: Technical Manual 116

13.12 Appendix L: Experiment: Statistical Tests 117

13.13 Appendix M: Experiment: Individual Crowdworker Results 126

Table of figures

Figure 1: IS Research Framework for a crowdsourcing platform. Based on Hevner, March, Park and Ram (2004) 14
Figure 2: Literature review 18
Figure 3: An example of a reCAPTCHA 20
Figure 4: Translation Quality Index with quality scale (Schiaffino & Zearo, 2006). 30
Figure 5: Active Crowd Translation (ACT) Framework 31
Figure 6: A sample of Urdu to English translation (Zaidan & Callison-burch, 2011) 32
Figure 7: Find-Fix-Verify pattern (Bernstein et al, 2010) 33
Figure 8: Current translation workflow for large enterprises, without the use of a crowdsourcing platform 35
Figure 9: Proposed translation workflow using a crowdtranslation platform 37
Figure 10: Action-Verification-Improvement (AVI) Unit 39
Figure 11: Introduction of two AVI units to resemble the Post Edition and Reviewing phase in a translation environment 42
Figure 12: Costs in cent per word, set off against the different user ranks 47
Figure 13: MVC architecture 48
Figure 14: Website interface 49
Figure 15: Post Editor and Reviewer Interface 49
Figure 16: Verification Interface 50
Figure 17: Post Editor and Reviewing Improvement Interface 50
Figure 18: Points of measurement to evaluate the working of the AVI-unit 51
Figure 19: Total Error Points and the respective verbal quality scale (between 'Reject' and 'Excellent') for a 350 word text (Schiaffino & Zearo, 2006). 53
Figure 20: Conceptual model for hypothesis 1 55
Figure 21: Conceptual model for hypotheses 2 and 3 56
Figure 22: Conceptual model for hypotheses 4 and 5 56
Figure 23: A table of confusion 59
Figure 24: Experiment design using crowdworkers with three different quality ranks (A,B and C). The experiment design is identical for both the Catalan and the Spanish experiment. 60
Figure 25: Histogram of the normalized translation test scores 63
Figure 26: Post Edition and Post Edition Improvement frequency diagrams and normal distribution plot 72
Figure 27: Total Error Points and the respective verbal quality scale (between Reject and Excellent) for a 350 word text (Schiaffino & Zearo, 2006). 73
Figure 28: TQI scores and the quality of that text (Schiaffino & Zearo, 2006). 109
Figure 29: Quality indexes for a sample user 112
Figure 30: Registration test, question section 1 115
Figure 31: Registration test, example section 2 115
Figure 32: Registration test, example section 3 115

1

Table of equations

Equation 1: Calculating the Total Error Points. 58
Equation 2: Calculating the Fixing efficiency 58
Equation 3: Calculating the Fixing efficiency 58
Equation 4: Calculating the Precision metric 59
Equation 5: Calculating the Recall metric 59
Equation 6: Calculating the Specificity metric 59
Equation 7: Calculating the Verifier Success Rate 59
Equation 8: Calculating the Translation Quality Index 106
Equation 9: Calculating the Error Allowance rate 106
Equation 10: Calculating the Total Error Points for the TQI 107
Equation 11: Calculating the Verifier Quality Index 111

Table of used tables

Table 1: Meetings held to discuss and evaluate the method and system. 17
Table 2: Open source community motivation taxonomy 24
Table 3: Translation judgment criteria 40
Table 4: Detailed explanation of the Post Edition and Reviewing AVI units. 43
Table 5: Error types and severity scores 44
Table 6: Levels of measurement 56
Table 7: Phase description for a single language experiment 61
Table 8: Normalized test results for the Spanish language test 64
Table 9: Normalized test results for the Catalan language test 64
Table 10: Independent two-samples t-test for the different translation test sections 65
Table 11: Proportion of crowdworkers and their expertise related to linguistics 65
Table 12: Number of users and their respective ranks 66
Table 13: Characteristics of crowdworkers 67
Table 14: Test for Normality using Shapiro-Wilk (Shapiro & Wilk, 1965) 68
Table 15: Outliers according to Grubbs' test for outliers (Grubbs, 1969) 68
Table 16: Results two sampled t-test for equality of means between Spanish and Catalan users 69
Table 17: Total Error Points found after the Post Edition phase 71
Table 18: Total Error Points found after the Post Edition Improvement phase 71
Table 19: Quality results for the different groups separate and combined 73
Table 20: Overview of errors with a high occurrence 75
Table 21: Quality results for the different groups separate and combined excluding the error categories Terminology and Style 77
Table 22: Binary classification metrics for crowdsourcing verifiers 78
Table 23: Binary classification metrics and their descriptions 78
Table 24: Original Sentence and Translation 80
Table 25: Errors identified by verifiers 80
Table 26: Efficiency scores for the different verification groups 82
Table 27: Summary of the results 83
Table 28: Summary of the results 84
Table 29: Summary of the results 84
Table 30: Error Categories, Types and Severity Percentages to calculate the TQI 107

1 Introduction

1.1 Introduction

Since the technological breakthroughs in the nineteen nineties the competitive playing field for businesses has been changing rapidly. Traditionally, their operating boundaries were limited by geographies, time and language. Now, due to the advances in communication networks and the invention of the internet this has been affected in a major way. This enabled organizations to expand their businesses globally, increasing their market size and customer base, while being subject to increased competition (Friedman, 2005).

To compete in this market, software giants such as Microsoft, SAP, CA Technologies and Symantec have to make software globally available and invest heavily in adapting their products to local standards and culture. This process is called the *localization* of a product and it involves the translation of user interfaces, marketing materials and user manuals. The market for localization, driven by globalization, has been growing steadily over the past years and has reached $30 billion in 2010 (CommonSenseAdvisory, 2011) and is expected to grow to nearly $40 billion by 2014.

The teams dealing with localization, translate English to numerous languages and hundreds of professionals are employed full-time to translate, manage and coordinate these processes. The costs are significant, running into 10's of millions of dollars each year per organization (CommonSenseAdvisory, 2011). Moreover, there are a number of limitations to the current localization processes:

- **Long time-to-market periods for non-English versions**: Localization processes are very time consuming. Products translated to non-English languages are usually released three months after the English version.
- **Unscalable workforce**: The translation workload is heterogeneous and there are some peaks during the year when a large number of products are released simultaneously. Due to these changing workloads, the localization teams are forced to outsource part of the software localization when workload is high, and have no work when the workload is low.
- **High cost of software localization**: Organizations invests several million dollars in localization per year both in internal and outsourced localization. This causes several products not to be considered for internationalization even for common languages such as Spanish, German or French.

- **High cost of extending to countries not currently in the translation portfolio**: Due to the high cost of localization it is unfeasible for companies to hire a team of translators for every language, especially for countries where expectations for sales are relatively low. This limits product exposure to numerous countries.

These limitations have been around for a number of years and have been difficult to solve without significantly increasing costs. Now however, a new paradigm could possibly solve these problems by using the collaborative power of the crowd. This paradigm, called crowdsourcing is the "*act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an* undefined, *generally large group of people in the form of an open call*" (Howe, 2006). Crowdsourcing can reduce the time to market and address the workload management through its scalability. Since virtually anyone connected to the internet and with knowledge of these languages can participate, the potential workforce is very large. The high cost of extending to new languages can therefore be resolved by crowd workers of those specific languages, while not being subject to the cost of these workers when no work is available. In addition, by applying crowdsourcing, costs for facilities, management and infrastructure can be leveraged, reducing the overhead of the localization process.

Unfortunately, many crowdsourcing solution are associated with low quality work (Adda & Cohen, 2011; Wais et al., 2010). The open nature of crowdsourcing leads to uncertainty whether the worker currently on the task is qualified to perform it well. Even when workers are selected based on certain criteria the workplace still holds eager beavers (identifying problems which are not relevant), lazy workers, scamming workers and slow learning workers. Identifying what work is good is a complex undertaking. It is impossible to have human reviewers from the organization verify the work, as the scalability and cost benefits would disappear. Creating automatic verification means for smaller tasks, less complex tasks, are a possibility, but not for more complicated and subjective tasks such as translations. The key problem is therefore to find a suitable means for quality assurance and apply this to create professional quality translations.

In this thesis an architecture to deliver crowd based software localization is proposed using the design science research methodology, the architecture is evaluated through an experiment. This exploratory research shows how organizations should explore the use of crowdsourcing

as a fast, cost-efficient and reliable alternative to business process outsourcing. Crowdsourcing provides elastic, on-demand, low cost human workforce in the same way as cloud computing provides computation resources. To exploit the potential of hundreds of thousands of multilingual speakers around the world, we propose an approach that draws concepts from distributed computing and previous work in crowdsourcing frameworks. We apply these concepts to improve current localization processes. The research question for this work is therefore states as:

"What translation quality can different configurations of crowdworkers deliver by applying a crowdsourcing quality assurance mechanism for software localization?"

1.2 Relevance

The term *crowdsourcing* has been a buzzword for several years. A number of successful crowdsourcing cases exist, but a real industrial application is yet to be created. In this thesis we will find out what it takes to crowdsource an industrial process. The scientific relevance lies in 1) the construction of a quality assurance mechanism for crowd work and 2) extensive results on the translation quality crowdworkers can deliver. Measured by professional translators instead of automatic tools.

The business relevance is that we find new ways to cut costs and improve the efficiency of processes. Crowdsourcing can become the next form of doing labor, not in an office, but from anywhere in the world in the comfort of your own home. By investigating whether an entire companies' process can be crowdsourced we set the standard for future crowdsourcing applications.

1.3 Deliverables

The research project leads to the following deliverables:

- Ranking algorithms and improvement of existing quality control mechanisms.
- A crowdsourcing prototype for localization purposes.
- Experiment results evaluated by professional translators.

1.4 Principal

The research described in this thesis was carried out at CA Labs, Barcelona. CA Labs is the research institute of CA Technologies, a global software company that provides distributed information technology infrastructure and cloud services. CA's localization department has an operating budget of over $40 million dollar per year and is looking into new methods to improve its efficiency and decrease its cost. The topic was proposed by Victor Muntes (Director CA Research Europe) and Patricia Paladini Adell (Vice President European Localization Services). To help build the prototype system two students of Universitat Politècnica de Catalunya (UPC) were involved.

1.5 Structure of Document

First, the research method is discussed in chapter two. Followed by the theoretical framework in chapter three. In chapter four we propose the first business processes crowdsourcing architecture for localization activities. Chapter five includes a detailed research method on the multilingual translation experiment. The sixth chapter describes the analysis of the experiment, followed by the results. The thesis concludes with a discussion, conclusion and section for future research.

2 Research Method

2.1 Introduction

The goal for this study lies in the exploration whether crowdworkers can deliver high quality translations and the evaluation of a quality assurance method for crowdsourcing software localization processes. The research design follows the design science theory (March & Smith, 1995) by using two design processes, namely the *build* followed by the *evaluation* of an artifact, also referred to as the *instantiation.* These steps have been placed in a comprehensive framework by Hevner et. al's (2004) Information Systems Research Framework for Design Science. The IS research framework for this research is illustrated in figure 1.

In the following subchapters, each of the factors in the research framework of figure 1 is addressed. First the *environment*, which dictates the business needs is described. Followed by a description of the *knowledge base* of the research, that addresses which methodologies and foundations are used to conduct the research. Finally, the development of the prototype and the evaluation method is described in the subchapter *IS research*.

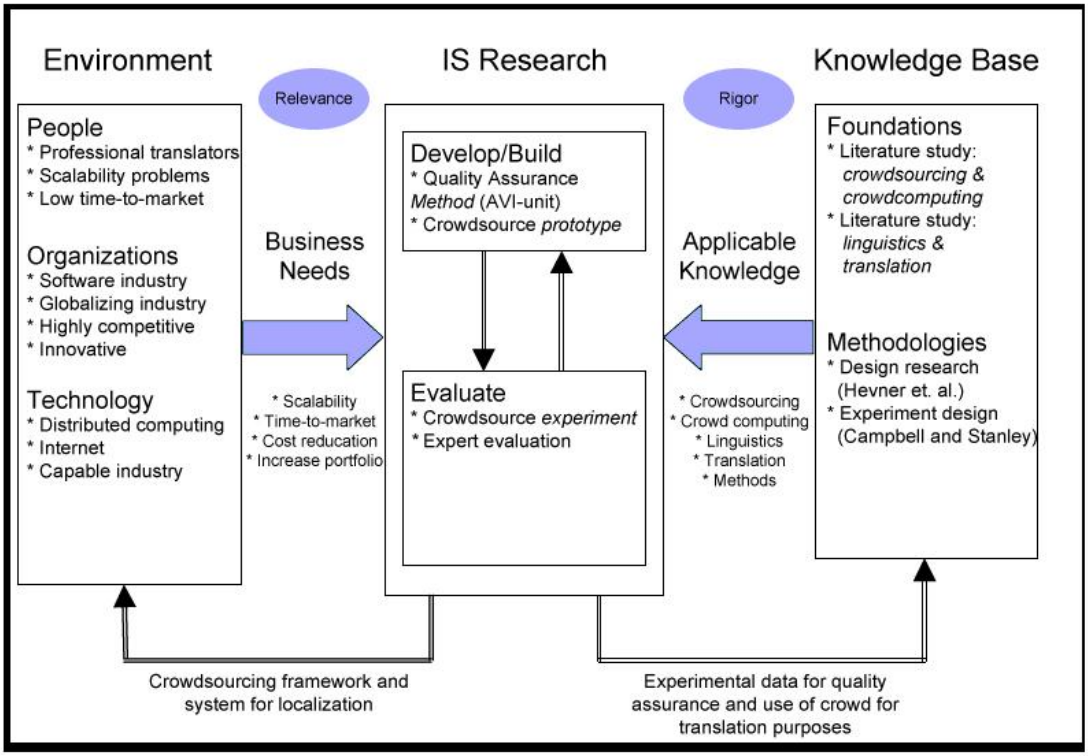


Figure 1: IS Research Framework for a crowdsourcing platform. Based on Hevner, March, Park and Ram (2004)

2.2 Environment

The *environment* encompasses the as-is and to-be situation of the business. The business needs are shaped by the capabilities and characteristics of the people. The context of the organization is used to assess these needs. Existing technologies and development capabilities are used to refine the business needs to a better product or process need. The resulting business need is to develop a platform to address four key problems in the localization services of large software companies.

- Long time-to-market for non-English versions.
- Unscalable workforce .
- High cost of software localization.
- High cost of extending to countries not currently in the translation portfolio.

2.3 Knowledge Base

The *knowledge base* is the foundations and methodologies required for the creation of the artifact and its evaluation. This step focuses on using theories, frameworks, models and methods from previous research to provide the means to build a proper artifact. The methodologies are used to evaluate the artifact through case studies, experiments and others. This research finds its foundations in two fields of theory, *crowdsourcing* and *crowd computing*. Drawing on an extensive overview of successful crowdsourcing platforms and methods on how to deal with its major weakness; quality assurance. By investigating the true motives behind the crowd worker and payment methods, the best reward system can be identified. The second part focuses on the field of linguistics and translations. Providing valuable insights into the difficulty of translating and the importance of components such as *lexicon, semantics* and *syntax* (Fromkin, Rodman, & Hyams, 1998). Current research into the translation industry provides examples of using the crowd in this context, and provide foundations to shape the platform.

Throughout the research, a number of methodologies are used to build and evaluate the proposed crowdsourcing system. The design methodology is based on Hevner et. al (2004) and the experiment design on Campbell and Stanley (Campbell & Stanley, 1963).

2.4 IS Research

2.4.1 Design and Build

Development of the artifact coincides with the development of a number of algorithms and processes that will support the automation of the translation process. The development is based on the knowledge base.

The *crowdtranslation platform's* (the artifact) goal is to provide high quality translations in multiple languages. The platform includes a quality assurance mechanism based on a literature study on crowdsourcing systems and meetings with experts. This will allow for increased scalability through heterogeneous workloads. Also the system allows exotic languages to be included in the translation portfolio. By automating the post editing process through the platform the business can cut managerial costs and even leverage the cost spent on the localization itself. From the worker perspective, the platform provides a flexible business partner where a decent salary can be earned and the worker can grow as a translator.

2.4.2 Evaluate

The evaluation of the design is split in two parts. First, the architecture and design principles are evaluated through meetings with professionals from the translation industry and the software engineering industry. Second, we evaluate the capabilities of the system through an online experiment with our prototype crowdtranslation platform.

2.4.2.1 *Design evaluation*

For the evaluation, we worked in close collaboration with a group of people from the research and translation industry. In addition, we planned two separate meetings with professional translators and software engineers to evaluate the design of the concept and the design of the systems architecture. Finally, we had periodic meetings with three experts in the field of research, computer science and machine translation to discuss concepts and methods.

Offline meetings were held at the CA office in Barcelona. Online meetings were held using CA’s conference system and involved a separate chair to mediate between discussions. For each meeting an agenda was made, relevant materials were sent to all participants at least a day in advance and relevant notes and actions were documented.

Table 1: Meetings held to discuss and evaluate the method and system.

Subject	Occurance	Duration	Participants
During bi-weekly research group meetings the progress of the project was discussed. The director of CA research Europe (Victor) and the Vice President of localization Europe (Patricia) would provide input and feedback.	Bi-weekly	1 hour	• Victor Muntes • Patricia Paladini Adell
Two sessions were held with two experienced translators of the CA localization team. Both translators had over 8 years experience within translation and provided feedback on translation standards.	2 sessions	1.5 hours	• Ilaria Cusumano • Clemens Bieg
A conference call was held with senior architects of CA technologies. These software architects provided feedback on the proposed method and design approaches for the development of the system.	1 session	1.5 hours	• Glenn Crossman • John Kane • Michael Stricklen
Marc (CA research), Joseph (Professor UPC) and Arafat (Machine Translation Expert CA) provided valuable insights on the method and means of evaluation and experiment setup.	Periodically	1 hour	• Marc Sole • Joseph Larriba • Arafat Ahsan

2.4.2.2 *Experiment evaluation*

The experiment is designed to answer questions regarding the quality the crowdsourcing platform can deliver. The experiment participants include volunteers ranging from profession translators to translation students to people with no knowledge of the translation or IT industry at all. All participants are ranked in different categories through a skill test. These rankings are used to prevent blocking and to run different types of experiments on the crowdtranslation platform.

The quality and improvement between the phases are calculated by using an assessment method called the LISA Quality Assurance methodology (Localization Industry Standards Association, 2003). Professional translators will determine the errors of the translations for each step within the experiment. These errors are then translated into Total Error Points (TEP), a weighted determinant for quality. The TEPs are also placed onto a more readable scale of quality. These metrics will present us with the means to identify the overall improvement in quality. Chapter 5 discusses the experiment in detail.

3 Theoretical Framework

3.1 Introduction

In this chapter, the theoretical framework for crowdsourcing and linguistics is given. Figure 2 describes the three sections and the underlying subjects that are addressed. The first section discusses the taxonomy of crowdsourcing, crowd computing and its current applications. This is followed by discussing the intrinsic and extrinsic factors that influence crowd workers' motivation. The second section discusses linguistics, taking into account the role of grammar, lexicon, semantics and syntactic components. This is followed by addressing the translation industry and the important role Machine Translation (MT) has claimed in the past 30 years. The last subject discusses the difficulty of evaluating the quality of a text and a translation. The last section addresses research crowdsourcing and linguistics research projects. We conclude by discussing the gap in the current approaches in quality assurance techniques.

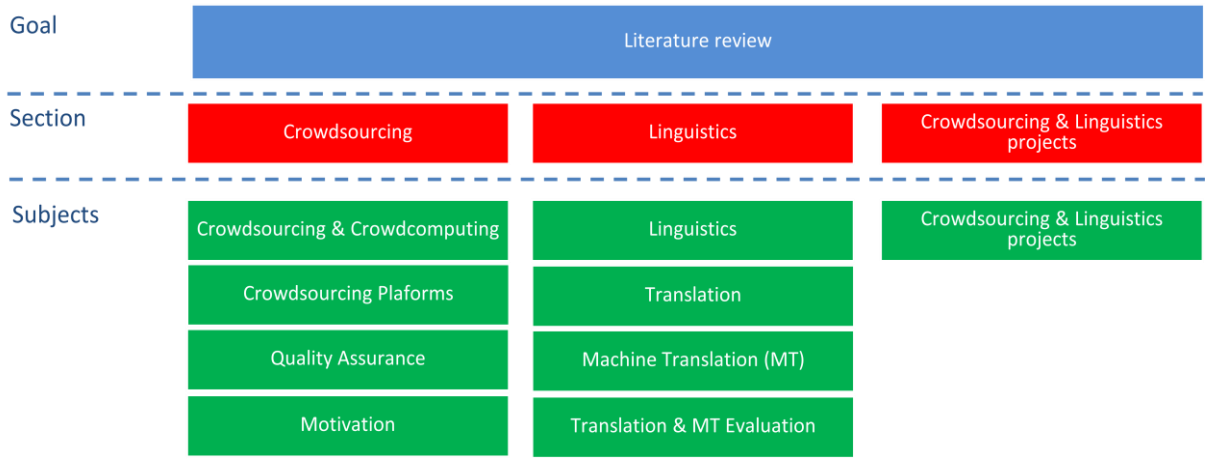


Figure 2: Literature review

3.2 Crowdsourcing

3.2.1 Crowdsourcing, Crowdcomputing and Human-based-computing

For a good understanding of what is meant by *crowdsourcing* it is important to look at a number of definitions. Friedman was one of the first to describe this phenomena and named it "*uploading*" (Friedman, 2005). Friedman emphasizes the combination of work performed by users that upload content from a personal computer to a network, encompassing the idea of crowdsourcing, but also that of open sourcing. Howe's definition from wired magazine (Howe, 2006) included the similarities with outsourcing and defines clear boundaries. He defines crowdsourcing as: "*the act of taking a job*

traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call". He emphasizes the term of an *undefined* crowd, usually through the internet, to complete work previously performed by a paid professional. Howe's definition quickly grew in popularity and has since been used for many online crowdsourcing initiatives. In 2011 Brabham expressed his dissatisfaction with the liberal use of the term crowdsourcing, arguing the incorrect use of the term for a wide variety of online and social platforms, not strictly meeting the five requirements set by Howe (Brabham, 2011). Brabham in response coined the following definition "*Crowdsourcing is an online distributed problem solving and production model that leverages the collective intelligence of online communities for specific management goals"*. His definition emphasizes the importance of completing *specific management goals*. This difference is important when making a clear division between crowdsourcing and for example, *open source,* where multiple software engineers at different locations develop, improve and share their sourcecode under the ten restrictions set by the Open Source Initiative (OSI) (OpenSourceAlliance, 2012). Brabham also argues that the difference lies in the management of the process (Brabham, 2008). As opposed to an open source initiative, where the management and control of the product lies with the community, in crowdsourcing platforms it lies with an external party seeking the *specific management goals*. In 2012, researchers acknowledged Brabham and his claims of the liberal use of the term crowdsourcing and performed an extensive literature study to try and determine an improved definition (Estelles-Arolas & Gonzalez-Ladron-de-Guevara, 2012). Their definition: "*Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage that what the user has brought to the venture, whose form will depend on the type of activity undertaken."* although comprehensive, is very complicated, and it does not significantly differ from those formed by Brabham and Howe.

In the wake of the term *crowdsourcing* lie two related concepts. First the term *crowd computing* and second the term *human-based-computation*. Each of these terms are closely related, but different to *crowdsourcing.* Examining the definitions more closely is important to be able to grasp the thin boundaries between them.

The concept of *crowd computing* is to achieve large-scale distributed computation through opportunistic networks (Murray et al., 2010). A good example of crowd computing was sketched by Anderson (2004) in a study where he shows how the crowd provides free computational power for a scientific study. Here any user could sign up and spare free CPU cycles to help on large calculations, while using the remaining CPU power for his or her own purposes. Although crowd computing uses the crowd to solve problems, it resides in a different segment then crowdsourcing. For example it does not leverage the human intelligence like the definitions of Howe and Brabham state.

Von Ahn coined the following definition for *human-based-computation* "*a paradigm for utilizing human processing power to solve problems that computers cannot yet solve*" (von Ahn, 2005). This concept is popularized through for example reCAPTCHA (Luis von Ahn, Maurer, McMillen, Abraham, & Blum, 2008). ReCAPTCHA is a service used by websites to determine whether a user is a computer or a human being, essential to provide security for a website. Figure 3 shows the essence of the program, where a user is required to enter the words 'about' and 'nghankn' to identify himself to the website. The human-based-computing part lies in the fact that the service also helps to digitize old books, newspapers and radio shows. In this example the word 'about' has been selected as part of a scan of a book. The computer was unable to transcribe the word and entered it in the reCAPTCHA engine. Whenever a user would verify his or her identity using reCAPTCHA he or she would help translate this segment of the book.

Figure 3: An example of a reCAPTCHA

The concepts of crowd computing, human based computing and crowdsourcing are similar in how they make use of the internet as a method of identifying potential parties. Each provides

a means to use external sources to solve problems. The difference of the concepts lies in the level of human involvement of the task. Where in crowd computing only external computational power is required, in human-based-computing and crowdsourcing human intelligence is required to solve the problem. Their differences lie in the fact that human-based-computation uses humans in combination with computational power, whereas crowdsourcing has humans perform tasks performed by humans before, but now sources that same task to a global market.

3.2.2 Crowdsourcing Platforms

In recent years a number of successful crowdsourcing initiatives have seen the light of day. Good examples are Threadless[1], the online t-shirt shop where users create their own t-shirt designs or iStockPhoto[2], the website that turned the stock photo business into an accessible business for amateur photographers to earn money and allows users to buy cheap and professional looking stock photos. Crowdsourcing has even reached the corporate R&D department through platforms such as Innocentive[3], where companies like Proctor and Gamble use the crowd to supply them with new ideas for customer products. Another example can be found in the case of GoldCorp (Tapscott & Williams, 2007) a large gold mining corporation which used over 1.000 prospectors worldwide to find the location of a valuable gold reserve. The results showed that using crowdsourcing gold veins could be found, which were deemed unfindable by professional geologists. uTest[4] employs the power of the crowd to provide huge corporations such as; Microsoft, Google and Amazon with testers for their systems. Their services include, functionality, security, load, localization and usability testing. They operate from 180 countries and offer over 50.000 testers. Crowdflower[5] is an enterprise crowdsourcing platform and delivers general purpose solutions. Crowdflower provides services like product categorization, content creation and many more by sourcing the work through its partners, and not to the crowd directly. The last, and perhaps most interesting platform, Mechanical Turk[6] gives businesses and researchers access to an on-demand, scalable workforce. Mechanical Turk is widely used to utilize human intelligence by posting tasks difficult for computers. The platform is therefore a combination of a

[1] http://www.threadless.com
[2] http://www.istockphoto.com
[3] http://www.innocentive.com
[4] http://www.utest.com
[5] http://www.crowdflower.com
[6] http://www.mechanicalturk.com

crowdsourcing and a human-based-computational workplace. The tasks, or HIT's (Human Intelligence Tasks), are posted on the online marketplace by the uploader. The workers, or Turkers as Amazon calls them, can select all tasks that fit their requirements. Tasks performed by the Turker are usually simple and self contained, such as identifying an object in a photo or selecting relevant search results, but can also be more complicated like translating a piece of text or writing a news article. The pay for the tasks range between $0.05 to $2.00 dollar per hour. The use of Mechanical Turk by researchers has grown steadily in recent years, who find the global and cheap workforce suitable for conducting experiments (Buhrmester, Kwang, & Gosling, 2011; Denkowski, Al-haj, & Lavie, 2010; Gao & Vogel, 2010; Negri, Bentivogli, & Marchetti, 2011; Zaidan & Callison-burch, 2011). Critics warn for the low wages and lack of rights for its workers (Silberman, Irani, & Ross, 2010). Stressing the need to address the social implications of working through these international platforms, where legal boundaries are still very immature.

3.2.3 Quality Assurance of crowdsourced work

In the past years crowdsourcing has been applied for many different types of tasks, for example to determine document relevance (Grady & Lease, 2010), find craters on stars (Leimeister, 2010), for content moderation (Kittur, Chi, & Suh, 2008) or translation (Zaidan & Callison-burch, 2011). Of all research performed, researchers agree that an extensive quality assurance method is required before data is usable. Platforms currently dealing with these needs for quality assurance have implemented different types of methods. Amazon's Mechanical Turk for example does not support its business customers or researchers with quality assurance. This leaves requesters (users or firms that publish the tasks) with two options, either to manually check for quality, or to post redundant tasks to crosscheck them for quality, an expensive and inefficient process. An experiment with the use of Turkers and workers from Crowdflower has been performed to reduce the work of translators (Negri et al., 2011), they identified how turkers can be used for translations but still required professionals to evaluate their work and correct mistakes. Manual quality assurance is expensive and removes the advantageous scalability of crowdsourcing. Redundant tasks are difficult to use in terms of tasks with subjective results (where multiple types of answers can be correct). Ambati, Vogel and Carbonell (2010) used redundant tasks in their experiment and yielded mediocre results, where in over 50% of the cases no agreement was found at all. Automatic means for evaluation seemed the logical next step. A number of researchers such as Ipeirotis (2010) have been working on crowd quality and have started working with the EM

(expectation-maximization) algorithm to identify malicious work (Dempser, Laird, & Rubin, 1977). These experiments are in its infancy, and have not yet been able to provide sufficient quality. Other approaches are proposed by CrowdFlower's researcher (Oleson et al., 2011) and Snow et al. (2008). They present a Gold-based quality assurance approach. A method where data with existing correct answers is inserted in the task list for the crowd workers. Whenever users provide an incorrect answer the user's accuracy rating will be adjusted and the user is trained by giving them feedback on the unit. This approach is supported by Kittur et al. (2008), who claim that verifiable questions greatly improve the quality assurance process. However, this is difficult when dealing with subjective tasks, where the complexity starts when trying to establish a (semi) automatic mechanism to monitor quality. How do we measure the degree of innovation of an idea, the beauty of a proposal or the best style for a text? These are measurements highly dependent on subjectivity, and in these scenario's human evaluation is a necessity. Bernstein et al. (2010) investigate the use of using crowdworkers to provide the quality assurance. By inviting multiple crowdworkers to vote on the best solution they identify which work passes the process and which does not. Unfortunately, in the final stage they rely on the requester to make the final decision of using this piece of work or leaving it out. Thus, combining the two verification techniques, using the crowd and manual checks. In addition the reviewers select only the best solution available, which does not imply that the quality is sufficient. The currently available quality assurance methods have been summarized in Table 1.

Table 1: Current quality assurance methods for crowdsourced work

Title	Advantage	Disadvantage	For example
Redundant Tasks	No interference with the system is required.	Costly and subjective tasks cannot be compared automatically. Results have been inconclusive.	Ambati, Vogel and Carbonell (2010)
Manual QA	Quality is guaranteed by having it analyzed by professionals.	This approach is costly, and prevents the crowd-application from being scalable.	Negri et al. (2011)
Gold-based approach	Provides the ability to check quality for specific tasks. In addition it is a good means for teaching the crowd on specific returning problems	Does not work well for subjective tasks and prevents the system from being able to provide ratings for the tasks not included in the gold approach.	Snow et al. (2008), Oleson et al. (2011), Kittur et al (2008)
Crowd QA	The crowdplatforms scalability remains. Less	Costs increases and the voting mechanism only	Bernstein et al.

	work is required from the crowdsourcer by providing him with the best available solutions.	selects the best options available, and does not identify whether it is correct or not. Manual QA still required.	(2010)

3.2.4 Crowdworkers and their motivation

How did Facebook, a multibillion-dollar company, manage its members to localize its website into multiple languages within weeks? Why do people perform tasks on Mechanical Turk for very little pay and what does the pay and type of task mean for the quality of the work? The research into motivational aspects of crowdsourcing is still in its infancy, and draws many aspects from research performed in online and open source communities. In an extensive literature study, von Krogh and von Hippel (2006) address the motivational aspects of contributors in open source communities. Kaufmann and Veit (2011) created a taxonomy divided into *intrinsic motivation* and *extrinsic motivation* (Table 2).

Table 2: Open source community motivation taxonomy

Intrinsic	Extrinsic
Self-determination	Concerns with competition
Competence	Evaluation
Task involvement	Recognition
Curiosity	Money
Enjoyment	Other tangible incentives
Interest	

Further research looks at the impact of motivation on the quantity and quality of the contributions. A number of researchers show how the intrinsic and extrinsic motivation of users and the type of rewards offered strongly affect the *quantity* of contributions (Fuller, Jawecki, Muhlbacher, Füller, & Mühlbacher, 2007; Kaufmann & Veit, 2011; Nov, 2007; Shah, 2006). All conclude how intrinsic factors are a strong motivator for increasing the number of contributions. Extrinsic motivators, like financial rewards, range from having a positive to a negative impact (Deci, Koestner, & Ryan, 2001; Eisenberger & Armeli, 1997). Borst (2010) sheds light on these mixed results by performing a follow-up research. She concludes how aligning the type of motivator to the motivation of the contributor is the key to

success. She states how extrinsically motivated people will increase contributions when offered financial rewards. People who are more intrinsically motivated do show an increase in contributions, but significantly less. In retrospect, studies on the influences of motivation on quality are less abundant. In a study performed by Rogstadius et al. (2011) the effect of intrinsic and extrinsic motivation on accuracy is tested. They deployed multiple experiments with different payouts, each repeated with two cover stories. The first story implicated that the worker would help a nonprofit health organization and the latter had them work for a profit organization. Results show how intrinsic motivation increased accuracy, but extrinsic motivation did not. However, the implication of this work is unclear. As other research states how extrinsic rewards work contraproductive for intrinsically motivated people (Charness et al., 2000; Heyman & Ariely, 2004). This emphasizes the importance of knowing who works on your platform, and what type of motivation fits your crowd best. Though, one thing is clear, reward systems are very important and its influence on the results is not to be underestimated. In addition, looking at just the crowdworker is not enough. Tapscott and Williams (2007) emphasize the importance of guaranteeing proper mutual rewards, for both the private firm and the contributor.

In a different study, Chandler and Kapelner (2010) show how meaningful work does not increase the quality of work. This contrast leads us to conclude that we cannot simply rely on the meaningfulness of the job at hand. The question remains whether linking financial rewards to performance improves the quality of work. Mason, Street and Watts (2009) show how increasing the reward does not necessarily increase the quality of the work, although the quantity of received work is influenced. However, their statement is disputable. In their research they rely solely on the Amazon Mechanical Turks policy of researchers being able to withhold payment if the task is not of high enough quality. An incentive to earn more by delivering higher quality was therefore not present. These objections are similar for Rogstadius et al. (2011) rewarding their workers regardless of the quality they delivered.

3.3 Linguistics

The study of linguistics is an important factor in the fields of translation and automated translation. Ever since the early 50's scientists have been trying to structure the rules of linguistics, and be able to create a machine translator that does not require human interference when translating text from language *a* to language *b*. However, these rules are so complicated for a machine to duplicate that they have not yet succeeded. To fully understand where the

difficulties of localization and the translation industry lie each of these subjects will be discussed in the sections below. We start by introducing the field of linguistics, followed by the field of translation and machine translation. The chapter ends by discussing the means to evaluate translations and machine translations.

3.3.1 Linguistics

Linguistics, has been a field of research for centuries (Lyons, 1981). The complexity of the linguistics field owes its roots to the meaning and structure of words in general, and is often described as grammar, or "*the basic units of meaning such as words, and the rules to combine all of these to form sentences of the desired meaning"* (Fromkin et al., 1998). Grammar represents linguistic competence, (Radford, Atkinson, Britain, Clahsen & Spencer, 2009) and can be divided into four key components, a (i) *lexicon*, a (ii) *syntactic*, a (iii) *phonological form* and a (iv) *logical form component* (Lyons, 1981). The lexicon- and syntactic components specify the dictionary of words and the course of combining them and forming sentences with correct semantics. Being knowledgeable in the language English would for example help you see that "*New York is bigger than Baltimore"* is grammatical, while "*New York is bigger then Baltimore*" is not, and "*New York bigger is than Baltimore"* is syntactically incorrect. The phonological form component includes "*the study of sound systems and processes affecting the way words are pronounced"* (Radford et al., 2009) (page 4) and is well explained by looking at their example in the process of *elision*, where in certain scenario's sounds can be dropped. For instance in the sentence "pint of milk" is pronounced as "pint o' milk", removing the F. This is however not the case when pronouncing "pint of ale", because ale starts with a vowel. The *logical form* component is resided with the interpretation of syntax and is linked to the difficulties of Machine Translation, which will be addressed shortly.

3.3.2 Translation

The field of translations is even more complicated due to multiplied ambiguity of multiple languages involved in the process. The translator is required to be knowledgeable in the lexicon, syntactic and logical form component for both. Even more complicated is the different syntax most languages use, increasing the difficulty of translating the text correctly. The essence of quality translations is well described by House (2001). She states how: "*the nature of the relationship between a source text and its translation text ... is essentially an operation in which the meaning of linguistic units is to be kept equivalent across languages*".

This means that in translation you are not dealing with the translation of the linguistic units alone, but even more about keeping the correct *semantics* of the sentence and text as a whole. In addition translating to different languages brings in a new set of challenges, related to culture or the way they describe the world. For example, in English we can distinguish blue colors, for example by mentioning dark blue or light blue. In Japanese they have words describe either the color green or blue, or somewhere in between.

3.3.3 Machine Translation

In recent years the use of software to provide translations between two natural languages has been booming. The research into Machine Translations (MT) is primarily found in linguistics as Computational Linguistics (CL) and in the field of computer science as Natural Language Processing (NLP) and involves the process using corpus and statistics to find the best way to not only translate the lexicon, but also convey the syntax and semantics (Jurafsky & Martin, 2008). There are numerous approaches to machine translation, well explained by Jurafsky & Martin (2008) in chapter 24. The field of Machine Translation owes its roots back to the formal language theory, widely used in mathematics and computer science as the basis to define a language in semantics and symbols (J. W. Backus et al., 1960; John W. Backus, 1959; Chomsky, 1956). The first serious idea of MT was proposed a couple of years earlier in an essay by Weaver (Weaver, 1949). A work that formed the basis for the first MT experiments. The field continued to shape through symbolic (abstract driven) and stochastic (probability driven) research (Jurafsky & Martin, 2008). In the 60's with the Brown corpus of American English a new step was taken into a still used phenomenon to use corpora for MT purposes (Hutchins, 1995). The corpus "*A collection of written or spoken material in machine-readable form, assembled for the purpose of studying linguistic structures"* (Merriam-Webster, 2008), contained over a million words and 500 written texts used for human language processing. Research teams from IBM managed to translate large portions of text correctly based on large bilingual corpora, convincing the community of its power. The field of CL and NLP adopted the use of corpora and slowly started to focus on statistical and symbolic models and started to shape the field of MT as we still know it today. Increasing popularity of word processors and personal computers usage saw the rising need for spelling checks. Commercial demand increased the interested in the fields and meant for a large increase in research being performed. To date MT is all around us through Microsoft Word's spelling checker, Google Translate, Babelfish and MT systems used by governments and companies. MT has proven to be a useful tool in recent years, but unfortunately has yet to

provide the world with high quality lexicon, syntax and semantics. Chomsky once presented the following problem with MT. Take the sentence "*I met a triangular man*", its syntax and lexicon are correct, but nevertheless it is quite unlikely to have ever happened. Critics argue how machines will never find the right or wrong in these sentences. A machine needs a mechanical rule to solve semantics problems, and these rules are as likely to provide correct or wrong conclusions (Bar-Hillel, 1963; Macklovitch, 1995; Madsen, 2009). Wilks (2009) analyzed these claims in his book on 40 years of MT research and to some extent supports these claims. He also notes however, how fast the field of Artificial Intelligence and MT have grown in recent years, and that we cannot be sure what the future will bring us. The quest for improving (machine) translations will continue, and an important aspect in this process is the aspect of evaluating its quality.

3.3.4 Evaluation of machine translation and translation

The 1960's marked the start for both the assessment of MT evaluation, and text quality evaluation (Jurafsky & Martin, 2008). For MT, Beebe-Center and Miller (1956) proposed two important concepts, one of measuring intelligibility and the other a method of measuring the edit distance by using known human translations. The Automatic Language Processing Advisory Committee (1966) conducted an influential experiment using trained human evaluators to score the quality of machine translations, using a likert scale the items fidelity and intelligibility were judged. These concepts proposed by Miller and Beebe-Center and ALPAC still provide important groundwork for example the extremely popular BLEU metrics, used by a large part of the research community to date. BLEU, invented by IBM (Papineni, Roukos, Ward, & Zhu, 2002) proposes an automatic metric that highly correlates with professional human translators. The concept of BLEU is based on n-grams[7] precision, comparing the n-grams of a candidate with the n-grams of a reference translation (also referred to as golden translation) and counting the number of matches. Related measures for evaluating machine translations are NIST by Doddington (2001) and METEOR by Banerjee and Lavie (2005). Other popular metrics are: Precision and Recall of Words (Melamed, Green, & Turian, 2003) and the Word error Rate based on the Levensthein distance. Measuring the quality merely by the use of automatic evaluation metrics however is in most

[7] N-gram is a statistical method in the computational linguistics field. The technique is based on the idea, if a letter x is found, what is the statistical likelihood of a next letter? For example, the chance the letter "b" is followed by an "a" = 0.3, b = 0.000002, etc.

cases not sufficient. The following example using string comparison shows how sentence (i) and sentence (ii) will score very high on similarity score, but mean the complete opposite.

> (i) "Though Tom was not sure why she said such a funny thing, he was going to take a train from New York City to Las Vegas yesterday."
> (ii) "Though Tom was not sure why she said such a funny thing, he was not going to take a train from New York City to Las Vegas yesterday."

The MT community therefore uses human evaluators to complement the machine translation. The most common method for this process is using a 5 point likert scale and measure the metrics Fluency[8] and Adequacy[9] of the translation (Denkowski & Lavie, 2010; Koehn & Monz, 2006). The downside however is the subjectivity that is brought into play when using human evaluators. Therefore, the reviewers need to have sufficient skills of both source and target languages. In addition to the earlier described difficulties of measuring machine translation quality, these mechanics have only been tested to work well on large datasets and with known reference texts to compare it too. When dealing with the evaluation of an unknown piece of text it is impossible without human translations, and we will have to turn to a different field of research. The field of judging translation quality is scattered in many directions. The first standardized scoring mechanisms have been developed by Kincaid (1975), by introducing the Flesch-Kincaid method and the automated readability index for the United States navy. These early models are based on simple techniques, using statistics to determine the quality of translations by using average number of vowels, syllables and lexicon difficulty. These types of measurements have spurred research into adding word frequencies (Elhadad & Sutaria, 2007) and syntactic complexity (Schwarm & Ostendorf, 2005). Pitler and Nenkova (2008) use human judgments to predict text quality, based on Likert rankings on subjects such as readability, how well the text was written and how interesting the text was.

In recent years businesses have been the driver behind the creation of standardized metrics to measure the quality of a translated text. Because, as W.T. Kelvin, known for determining the absolute value of zero, once noted "*if you can measure what you are speaking about and express it in numbers you know something about it; but when you cannot measure it, when*

[8] Fluency measures if a text is fluent, regardless of the meaning
[9] Adequacy measures if a text is conveying the meaning, regardless of the fluency

you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind", businesses feel the need to be able to judge the work in an objective way, and require a measurement of work to be able to improve it. This mindset has led to the creation of Translation Quality Assessment models such as SAE J2450 (Society of Automotive Engineers, 2000), an industry standard for categorizing errors in the automotive industry and the LISA QA (Localization Industry Standards Association, 2003) for measuring quality in a wider range of project types. For LISA QA, the most popular metric, errors are defined based on a list of 26 pre-defined error categories and errors get assigned a weight to indicate the severity of the mistake. Although LISA has been closed down since February 2011, many localization teams still build on the initial LISA QA system. Adapting the error categories, severity weights and calculation algorithms to their specific company or project QA needs. To give an idea of the quality based on these errors, Schiaffino and Zearo (2006) created the Translation Quality Index (TQI). A quantitative-based methodology for calculating the index of a text, based on the *errors* and *total number of words* of the translation. The TQI couples the score with a verbal quality scale (Reject to Excellent) (Figure 4: Translation Quality Index with) to give non-professionals an idea of the quality. Professionals usually score in the excellent range.

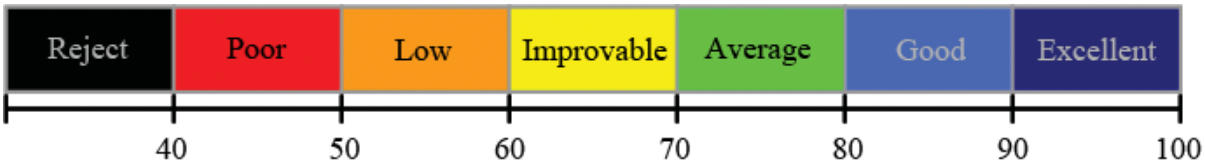


Figure 4: Translation Quality Index with quality scale (Schiaffino & Zearo, 2006).

Though these metrics have been helpful, whatever metric is used, researchers and professionals agree that judging the true quality of a text holds a certain portion of subjectivity and for now professional translators have been used to perform the final judgment.

3.4 Crowdsourcing and Linguistics

In recent years, researchers in NLP and MT have been exploring the use of crowdsourcing for improving machine translations and post edition purposes. Ambati, Vogel and Carbonell (2010) present the Active Crowd Translation (ACT) framework (figure 5) that combines active learning, the process where the MT system learns new words and sentences, and crowdsourcing. The ACT framework is designed to provide improved corpora for MT engines, and thus improving the quality of machine translations. As discussed in the section

on *quality assurance*, the ACT framework makes use of redundant tasks to determine which sentences are correct and wrong. The framework uses multiple crowd-translators to translate similar sentences and disposes of translations where translations are not similar. This technique resulted in over 50% of the work being removed. They recognize the difficulties laying ahead stating *"A more challenging task is to perform matching when there could be more than one perfectly valid translations for a given sentence."* (Ambati, Vogel & Carbonell, 2010). Unfortunately no real claims can be made based on their dataset as they used BLEU to evaluate the quality of the texts. A measure that is only proven to work on very large texts and only provides an indication for quality, but is no replacement for human evaluation (Papineni, Roukos, Ward, & Zhu, 2002).

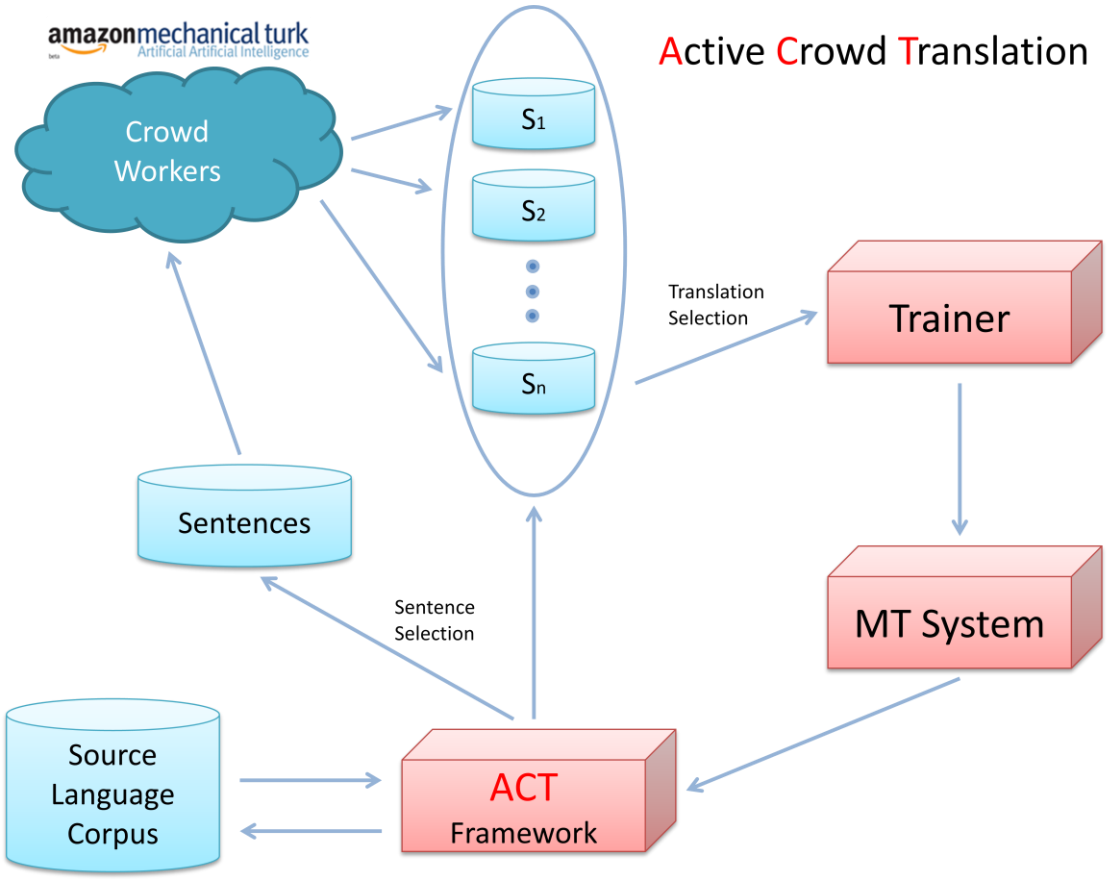


Figure 5: Active Crowd Translation (ACT) Framework

Zaidan and Callison-Burch use a similar approach to create high quality translations. In their paper *professional quality from non-professionals* they demonstrate how some translations resemble those of professionals (figure 6). The authors record a number of factors per translator to use in calculating a weighted endscore per translation. Factors include, mother tongue, word-error rates and edit rates from other translations. The total score is compared to MT evaluation scores. The system shows how crowdsourcing can result in decent quality

translations, but the majority did not reach quality requirements, nor is there a means to evaluate the quality.

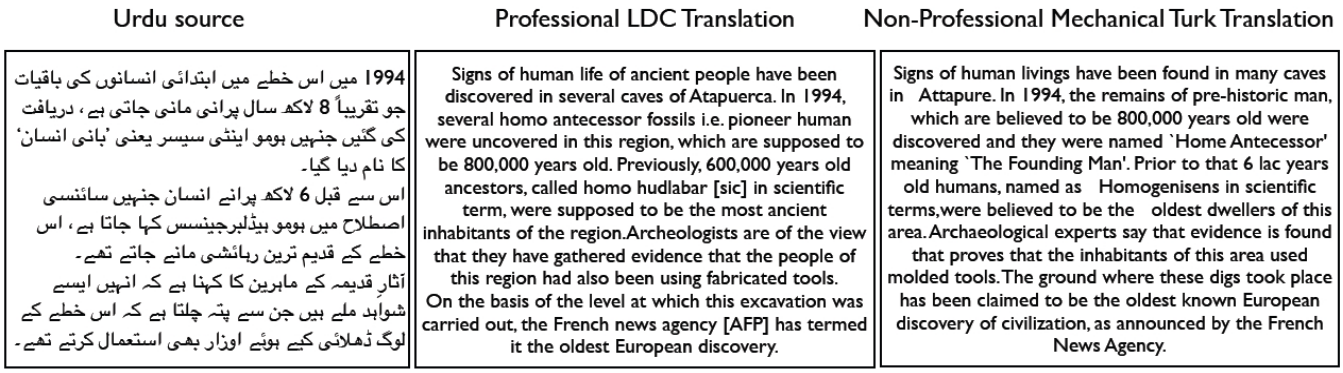


Figure 6: A sample of Urdu to English translation (Zaidan & Callison-burch, 2011)

In another research effort, Microsoft Research developed the Collaborative Translation Framework that utilizes their community as Post Editors. The framework is a combination of training the translation memory, use of machine translation and use of a wiki to post-edit translations. The novelty in the system lies in the use of the crowd to improve the text through a wiki-based website. Texts edited by the community can be ranked and improved by other members. The downside lies in the lack of quality assurance, as there is no feasible method to ensure quality for all the translated sentences. In response Microsoft Research proposes to use the crowd to provide proofreading and syntax improvement through Word and Mechanical Turk. In a more recent work in a combined research effort, the Solyent framework was proposed. Solyent, developed by Bernstein et al. (2010), is a crowdsourcing platform directly linked to the Word text editor. Users with the Soylent add-on would be able to use the Word interface to identify paragraphs or sentences and indicate a problem or question they have regarding the selected text. These problems, ranging from a request to find a suitable picture, change text from past to present tense or find the bibtex references would be automatically crowdsourced. Through the proposed Find-Fix-Verify pattern, they split the tasks into a series of stages that utilize independent agreement and voting to produce reliable results.

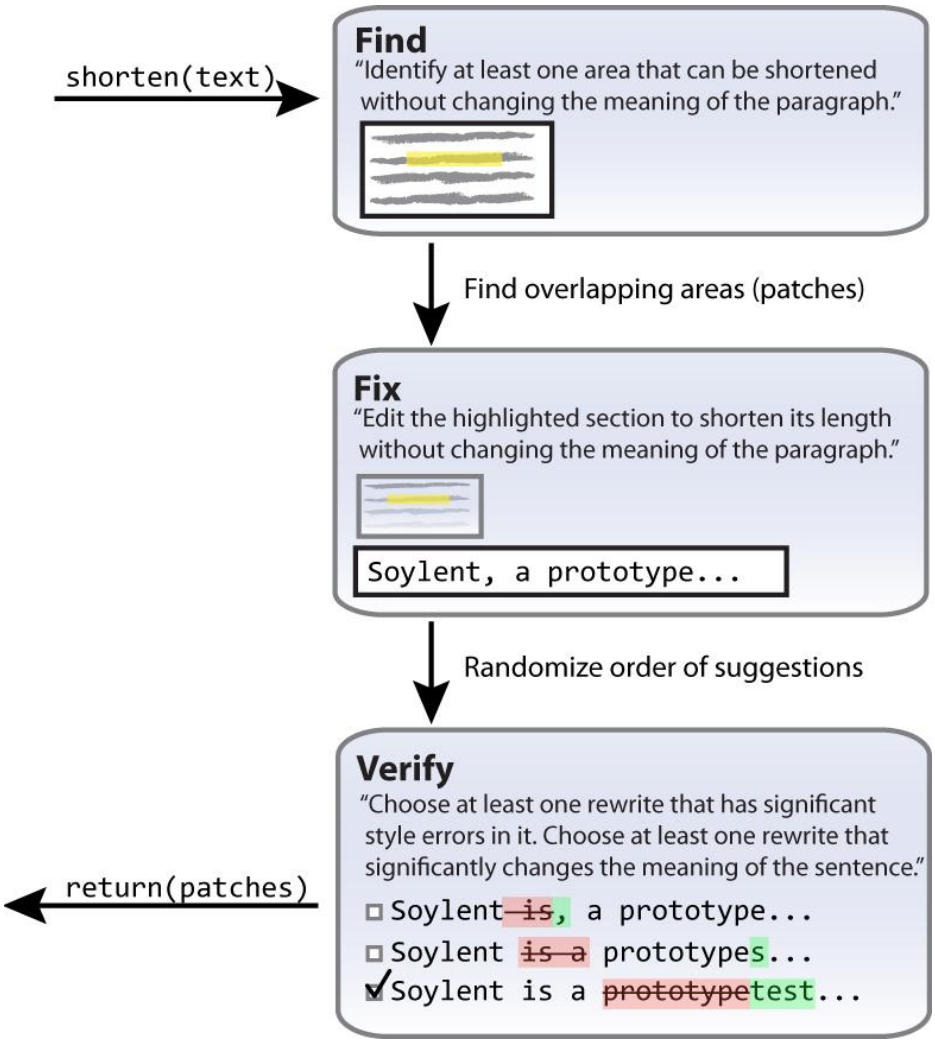


Figure 7: Find-Fix-Verify pattern (Bernstein et al, 2010)

he example of figure 7 indicates a request from a user to shorten a paragraph without changing the meaning. In the find stage multiple opinions from turkers are compared to only let those with high agreement rates on errors progress to the fix stage. In the fix stage 3 to 5 workers are recruited to fix the patches of interest identified in the find stage. In the verify stage the results are proposed to verifiers who select the best-proposed patch from the fix stage. The best results are sent back to the crowdsourcer who decides whether to accept or decline the changes made by the crowdworker. Through experiments Bernstein et al. (2010) found how 70% of the work was correct, but 30% would still contain errors. They argue how "lazy workers" and "eager beavers" cannot properly be identified and prevented from participating in the system. In addition, they discuss how lack of domain knowledge can jeopardize quality.

3.5 Gap in current Quality Assurance techniques

The use of crowdsourcing is quickly growing in popularity and researchers are experimenting using the crowd for more complex tasks. To deliver a high standard of quality QA mechanisms are a necessity, especially when judging subjective work such as translations. Researchers such as Ambati, Vogel and Carbonell (2010) and Bernstein et al. (2010) realize that when dealing with the quality assurance at least two things need to be kept in mind. First, automatic evaluation techniques such as BLEU (Papineni et al., 2002) and METEOR (Banerjee & Lavie, 2005) cannot be used in a live crowdsourcing platform. These methods rely on the existence of pre-existing golden translations, translations rarely available for unpopular languages and never available when dealing with new texts. Thus the evaluation has to be done by humans. Second, the QA process has to be performed by crowdworkers. This because a system where the crowdsourcer has to perform the check for quality erases the scalability benefits of crowdsourcing and it nullifies a large part of the savings that can be made, as employees will still have to manually evaluate all the work.

To address these concerns they introduce working with crowdworkers to check for quality. A promising new paradigm where they first let crowdworkers perform a certain task and then let others perform a verification. However, in the first tests, they have not been able to deliver the desired quality standards the market is looking for. Mainly because their systems do not actually *improve* the quality of the work, and only verifies which work is *best* from a certain selection of jobs. Receiving high quality work from this system will be therefore be highly dependent on the capabilities of the crowd workers. This means that when only low quality work is submitted, the best of the worst will get selected as the highest quality translation. Which is certainly not enough when high stakes are at play.

This gap in the QA process when working with crowdworkers is what this research aims to solve. We look to aim to not only verify which task is best, but verify the quality based on a number of quality metrics. In addition, we aim to use this feedback to actually improve the quality of the work. In addition we determine what impact the capability of the worker has on this process. Is it possible that low quality workers can deliver high quality translations if they collaborate with higher ranking co-crowdworkers? We propose a system where we do not only have the crowd perform a certain initial *action*, or first tasks. We also have crowdworkers "*verify*" the true quality of that work, and have them use that information to *'improve'* it afterwards. We name this new QA system an Action Verification Improvement unit (AVI-unit).

4 Crowdtranslation Platform

4.1 Introduction

In this section, we explain the creation of the crowdtranslation platform and the new mechanism to guarantee quality for crowdsourcing environments. The system we propose follows the same structure as the translations industry does, using a post edition and reviewing step for work. To address the quality concerns we build on the find-fix-verify pattern proposed by Bernstein et. al. and further introduce the AVI unit.

Chapter 4.2 addresses the translation workflow and how the crowdtranslation platform fits in this process. Chapter 4.3 introduces the AVI-unit, a QA system that combines multiple aspects of the different QA methodologies currently being used in crowdsourcing system (table 1), while benefiting from their advantages and mitigating their disadvantages. Lastly, chapter 4.4 addresses the technical architecture of the platform using the Model View Controller architecture and provides screenshots on the system.

4.2 Translation Workflow

Before we introduce the workings of the AVI-unit we describe the current translation workflow, and what part of this workflow will be crowdsourced in the proposed system.

4.2.1 Current Translation Workflow

Large enterprises employ localization teams with professional translators, which follow a certain process to deliver translations. The conventional process, applied by organizations worldwide, follows five consecutive steps, illustrated in Figure 8: .

Figure 8: Current translation workflow for large enterprises, without the use of a crowdsourcing platform

The steps work from a source text to a complete translated text.

1. A stripped version of the user guide is prepared for the localization workflow. The writers of technical manuals use strict formatting guidelines before entered in the system. These include referencing tags to identify the location of images and tables. These tags are included in the document and are identified by combinations of special characters, much like those in programming languages, for example: <image>. This document is divided into chunks or blocks.
2. The machine translator takes the source document and uses its dictionary to creates the best possible translation. This MT engine takes into account semantics, syntax and other linguistic components. These dictionaries are improved by previously published high quality work.
3. The target text will contain errors and will need further improvement.
4. A localization team of an organization will provide the *post edition* of the machine translated text. The document will be improved using a standard set of guidelines and follows strict rules using a glossary. A translator would normally be responsible for a number of sentences or pages. The localization team reviews the post edited text and improves it even further. A translator will be responsible for the *reviewing* of a post edited text of a different translator, and correct and improve it, where needed.
5. A completed translated text is ready to be styled or imported back into the application database.

4.2.2 Proposed Crowdsource Translation Workflow

The crowd-based localization system we propose follows a similar process as described in the previous chapter. The crowdsourced process differs as the post edition and reviewing process now take place in the crowd. The new workflow is illustrated in Figure 9.

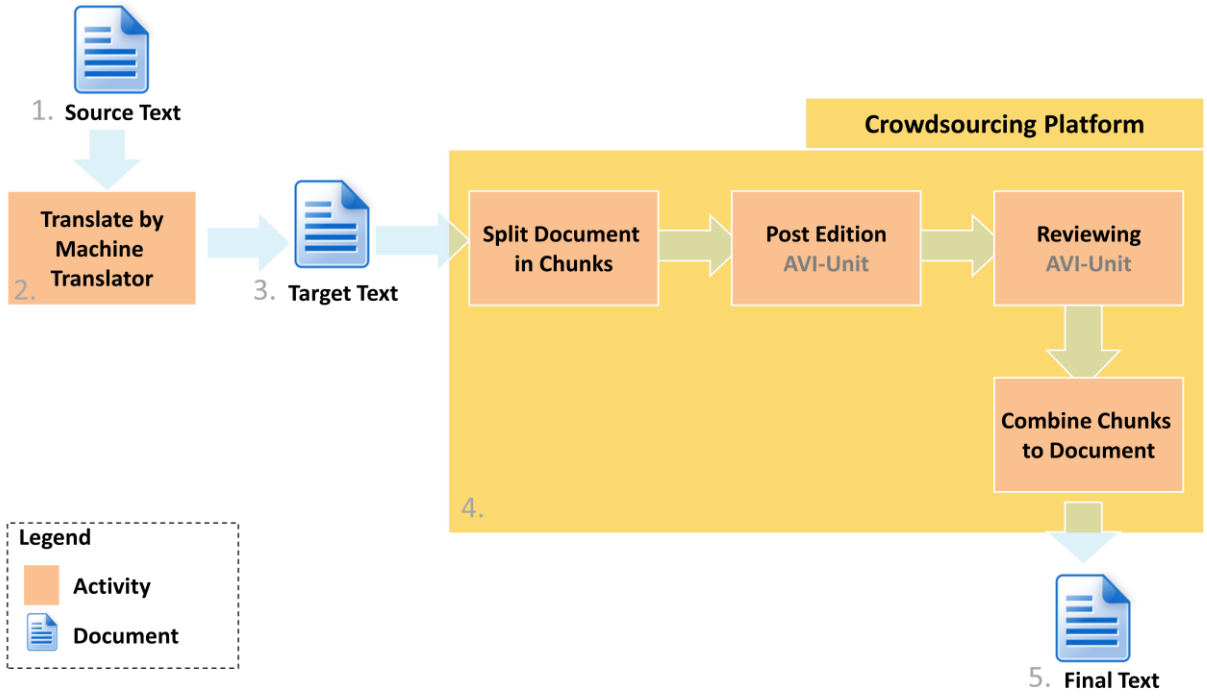


Figure 9: Proposed translation workflow using a crowdtranslation platform

4. In the crowdsourcing platform, instead of handing the target text over to the localization team after the text is translated by the machine translator, it will be inserted into the crowdsourcing platform. Here, the complete target text is split into multiple blocks by using MapReduce. The splitting in chunks is performed to decrease the task size for crowdworkers. This step uses the MapReduce technology which was first introduced by Malone and Crowston (1994).

   After the target text is split into chunks, the text will be sent to the post edition phase, where the work will be performed by crowdworkers through an AVI-unit. The workings of the AVI-unit are explained in chapter 4.3.

   As soon as the Post Edition phase is competed the reviewing step will commence, once again by crowdworkers through an AVI-unit.

   Once the crowdworkers have completed their texts they are automatically put back together by combining the chunks and delivering the final translated text.

4.3 A Quality Assurance mechanism for translation tasks

Four aspects characterize the design of the crowdsourcing system we propose. We introduce an AVI-unit to replace the professional translator in the post edition and reviewing phase in the translation workflow (figure 9). These AVI-unit makes use of quality assurance mechanics such as redundant tasks and crowd verification. We complement these techniques by adding an improvement phase, where the result of the verifiers is used to improve the work from the initial task. In addition, a new ranking algorithm for crowd workers is created. This allows us to determine the trustworthiness of workers and predict the probable quality of their work. Lastly we will use the ranking mechanism for a fair reward system, which rewards crowdworkers based on the quality and amount of work delivered. The following sections explain each aspect in detail.

4.3.1 AVI-Units for Quality Assurance

Having crowdworkers perform the steps of post edition and reviewing brings one key challenge into the equation: Quality. In a regular working environment the translation team works with professional translators with a proven track record. Also these employees have a strong connection to the organization they work for and their employment is dependent on them delivering quality. In a crowd environment this is no longer the case, here the organization has no or little control over the people who will be translating. This demands for new means of quality assurance to make sure the quality of a post edition or reviewing job is guaranteed.

To address these concerns we propose the AVI-unit for both the Post Edition and Reviewing phase of the translation workflow. The AVI-unit (Figure 10) is a co-creation three step quality assurance mechanism. In the AVI-unit one worker performs a certain action or task. After the task is completed, it is verified by one or more other crowdworkers. Where in the third step we combine the verifiers data and provide feedback to the initial worker. By congregating the feedback and providing this to the initial worker we establish a relationship pattern between the workers in the crowd to help them collaboratively improve the quality of translations. The key characteristics, such as redundant tasks and multiple step verification are derived from the Find-Fix-Verify pattern proposed by Bernstein et. al (2010) and the ACT-Framework by Ambati, Vogel and Carbonell (2010). We improved the model by adding the improvement step which allows for improvement based on feedback and learning for crowdworkers based on made mistakes.

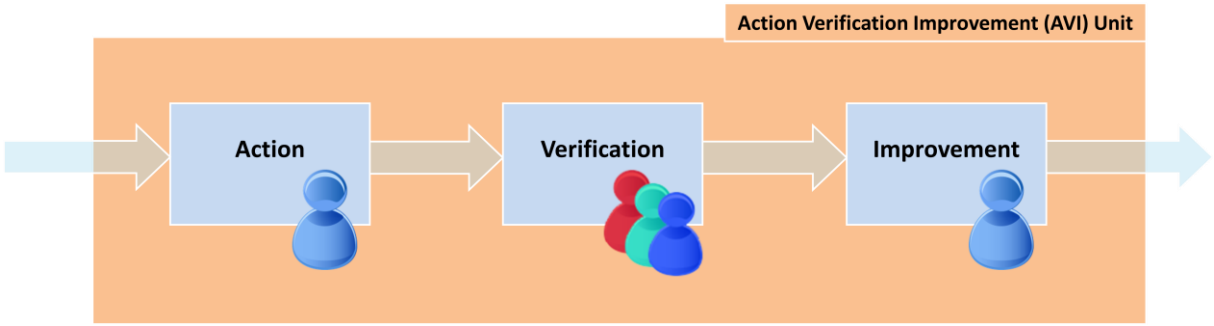


Figure 10: Action-Verification-Improvement (AVI) Unit

Action Phase

In the action phase the crowdworker performs a certain task or action. In the translation platform the task is presented as a number of small tasks in the form of individual sentences. For each sentence the original English sentence and the target language machine translation is provided. The crowdworker is instructed to correct the machine translation which is in line with grammatical, technical and company guidelines. We provide the worker with a stylesheet with these guidelines, a glossary of important terms and the original English pdf file of the text.

Verification Phase

In their work, Bernstein et. al (2010) have reached their quality standards in 70% of the reviewed cases. Ambati, Vogel and Carbonell (2010) reached high quality results measured in BLEU metrics. This indicates a certain quality level, but not prove professional quality. Professionals will require the texts to adhere to certain organizational standards and guidelines, which metrics such as BLEU will not be able to measure, in addition BLEU gives an indication of the quality, but can never be certain on certain style, grammatical and semantics issues. In their work they tried to determine the correctness of a sentence by comparing it to different users. Whenever a sentence would be identical it would be defined as correct. An approach unfeasible for subjective tasks such as translation, where sentences can differ but still be correct. In addition users can provide equal sentences but still be wrong. This method of implementing redundant tasks is interesting, however we feel it is better put to use in a verification phase, like Bernstein et. al (2010) propose. They however, use a popularity vote to identify which task is best. An approach which does not say anything about the actual quality itself. We propose a system where we use redundant tasks and *judgment*

criteria, specific to the task at hand, to specify which kind of problems exist in the task perform by the worker in the action phase. These criteria are then used to calculate conformance[10] between the verifiers and determine the actual *quality* of the work, not just which one of a number of works is *best*. For our translation platform this means identifying what type of linguistics problems are found in the translation.

To evaluate machine translations measures such as *adequacy* and *fluency* are used to determine quality. However, these are relatively simple and provide little information regarding the origins of the error. The professional translation industry uses a very complex measuring system to evaluate the quality of translation service providers. In their approach they periodically review a sample of their work and determine whether they continue or terminate contracts. This approach consists of 43 error types (appendix D) each with 4 severity levels (appendix E) and is very precise. Having the three crowdworkers participating in the verifiers phase identify the severity for every single sentence would be very time consuming. To address this concern we collaborated with CA professional translators to devise a simplified set of judgment criteria, each with the same severity level. This set has been based on a mapping between the error types of appendix D and is shown in appendix F. The resulting criteria used by verifiers are depicted in table 3.

Table 3: Translation judgment criteria

Error type	Description
Mistranslation	The target language does not accurately reflect the meaning of the source text. This may include ambiguously or literally translated passages if the meaning of the original text is lost or altered.
Omission/ Addition	Source text information has been deleted from the target text, or information, not found in the source text, has been added to the target text.
Inconsistent terminology	Inconsistent terminology errors are where the translation does not follow generally accepted industry or company standards.
Grammar	The translation does not adhere to the specific rules of the target language with regard to grammar.

[10] The algorithm behind the conformance calculation is described in chapter 6.3.3.2

Style	The translation does not adhere to company style guidelines and/or other specifications provided.
Punctuation	The translation does not adhere to the specific rules of the target language with regard to punctuation.
Software options	The user interface elements (names or menu options, windows or dialog boxes, etc.) used in the translation are not identical to the glossary or do not respect the capitalization as indicated in the guidelines.
Untranslatable text	Untranslatable variables (html tags, product names etc) or other content indicated in the guidelines is translated.
Typographical error	The word does not comply with specific language with regard to spelling. This error category includes typographical errors or misprints: unneeded spaces between words, omission of letters or order of the letters.

Improvement Phase

In the models proposed by Bernstein et. al (2010) and Ambati, Vogel and Carbonell (2010) the information gathered through redundant tasks and verifiers are only used to determine whether the quality is good or not, and valuable data, which could be used to improve the work is discarded. By introducing the improvement phase we gain two advantages: 1) we establish a pattern between workers where they collaboratively create a translation, and 2) we create a system which allows translators to learn from their mistakes, and improve their translation skills. To supplement the error identification the verifier provides feedback for the worker in the improvement phase. The user in the improvement phase is the same worker as in the action phase.

4.3.2 Post Edition and Reviewing combined with AVI-units

We will use the AVI-units to emulate the translation workflow as indicated in Figure 8. The classic quality control mechanism, of using a Post Editor and a reviewer, will be reused in the crowd environment. The crowdsourcing workflow as illustrated in Figure 11 introduces this concept where the first AVI-unit (Post Edition) creates an initial translation and the second AVI-unit (reviewing) reviews and improves this work further (this diagram only depicts the steps performed within the crowdsourcing platform). The steps of the of the complete crowdsourcing system are explained in table 4.

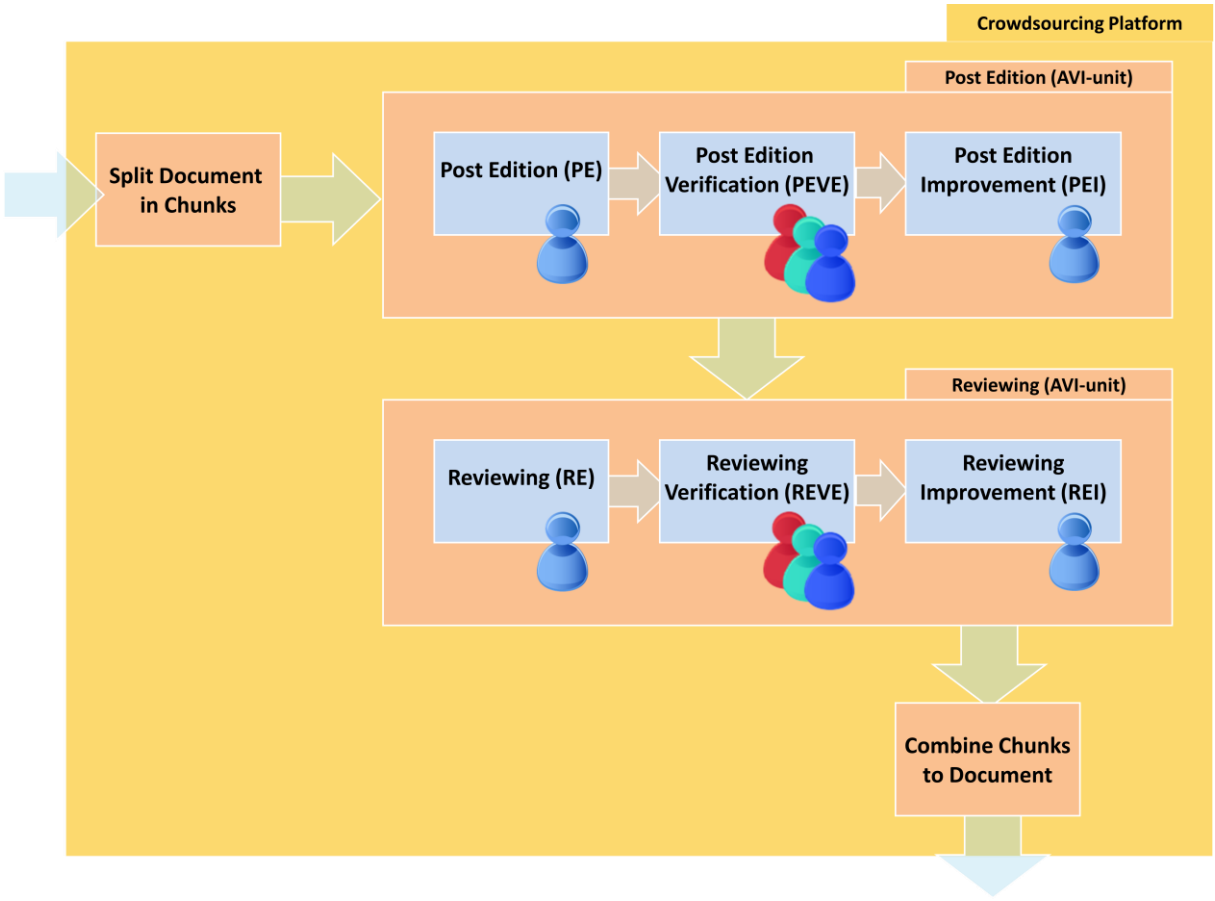


Figure 11: Introduction of two AVI units to resemble the Post Edition and Reviewing phase in a translation environment

Table 4: Detailed explanation of the Post Edition and Reviewing AVI units.

Post Edition AVI-Unit	In the post editing AVI-unit the first task is the post edition of the text (PE phase) by a bilingual user. The *Post Editor* will receive the original sentence in English, or source text (T_{source}) and the machine translated text in the target language, or target text (T_{target}). The Post Editor's task is to improve the machine translation to fluent target language sentences. In the post edition verification task (PEVE phase) three verifiers will provide the *Post Editor* with feedback on his or her work. Each of the *post edition verifiers* is selected based on ranking and feedback is compared to determine what information to return to the *Post Editor*. The verifiers are presented with T_{source} and the improved $T_{target.}$ To determine the quality of the sentences the verifiers are presented with the judgment criteria to identify erroneous sentences. A feedback textfield is included to provide suggestions regarding the problem. The judgment criteria are depicted in Table 3. After the verification task the *Post Editor* (PEI phase) will receive the feedback for erroneous sentences and improve his work. The work is then forwarded to the Reviewing AVI -Unit.
Reviewing AVI -Unit	In the reviewing AVI -unit the first task (RE phase) is to have a native speaker review the text delivered by the Post Editor. The reviewer is only presented with the T_{target} output from the Post edition AVI unit and reviews the text. Reviewers tasks do not include improving the translation, but focus solely on the semantics, syntax and lexicon aspects of the sentences. The second task in the reviewing AVI-Unit is the reviewing verification (REVE phase). Here two or more verifiers are selected to identify errors and provide suggestions for the work. The judgment criteria are similar to those of the PEVE phase, but error types such as *Omission/Addition* and *Mistranslation* are excluded. The reviewer (REI phase) improves his work based on the feedback of the REVE phase and submits the final text.

Besides using the added improvement phase we introduce the concept of worker-ranking to complement the AVI-units. We will use the work performed by crowdworkers (and their evaluations) to determine how well they perform and rank them accordingly. Based on their rank user decisions will receive added weight to their judgment, or require less verifiers to check their work.

4.3.3 Worker Ranking for Quality Assurance

To determine user trustworthiness, we calculate the quality of a users work based on the reviews of other crowdworkers, or how they compare to other verifiers. The rank a user has is also referred to as the Quality Index (QI) and builds on the TQI translation judgment method from Schiaffino and Zearo's (2006) by taking into account a simplified error identification system. To calculate the QI we propose two ranking algorithms of which the goal is to increase a users rank after completing high quality work, and to decrease the rank after providing low quality work. This QI is also used to determine the reward the workers receive per completed task. This chapter discusses the two different ranking mechanisms. In addition we discuss a method to fairly calculate average ranking of users through an exponential histogram.

4.3.3.1 *Ranking Indexes*

Post Editors and reviewers work is being ranked after each task they complete. If the job contains little errors the quality of the work will be judged with a high score, if the job contains many errors, it will receive a low score. To calculate the ranking index we use a modified version of the Translation Quality Index (TQI) which is further explained in appendix G. The TQI uses a system where certain errors carry more weight, or are counted to be more *severe* than others. Table 5 shows the errors, previously introduced in table 3, and their weight. This means that *mistranslation* errors are 2,5 times as severe as grammar errors.

Table 5: Error types and severity scores

Error Type	Error Weight Multiplier	Error Type	Error Weight Multiplier
Mistranslation	2,5	**Punctuation**	1
Omission/ Addition	1,5	**Software options**	1
Inconsistent terminology	1	**Untranslatable text**	1
Grammar	1	**Typographical error**	1
Style	1		

The scale for ranking Post Edition and reviewing *verifiers* is called the Verification Quality Index (VQI) and is based to what extent the verifier deviates and agrees with other workers verifying the same piece of work. In addition the VQI takes into account the current QI of the user and uses this to compare their work with other verifiers. The calculations for the VQI can be found in appendix H.

4.3.3.2 *Rank over Time*

In the previous sections we discussed how the quality of an individual task is ranked. In this section we discuss the best approach to use these ranks to determine the overall rank of the user. The most convenient approach would be to average the quality indexes of all the users tasks and use that as the overall rank. This would however discriminate early work and early performance would haunt the worker.

To resolve this problem we propose the use of an exponential histogram (Datar et. al, 2002). In an exponential histogram we use a differentiated weight of the work delivered based on the time the job was done. This approach is used to value work based over longer periods of time, without discriminating early work and taking into account improvements overtime. The workings of the exponential histogram is explained in detail in appendix I.

4.3.3.3 *Trial Period*

When new workers join the system, the first jobs are performed in trial mode. This allows the user to get used to the system without damaging his/her future rank and payout. After a predefined number of jobs have been completed the trial mode ends. When a users ranking index drops below the rank of 40 he or she is flagged by the system as being a low performer. After a certain amount of time without considerable improvements the user will not be able to take on new tasks.

4.4 Rewarding Mechanism

People's reasons to participate in a crowdsourcing system rely on intrinsic and/or extrinsic motivational factors. Whether they want to try to help a community, participate in a collaborative process to build something new or to be rewarded economically, differs greatly. However, the *crowd translation platform* is intended to replace an industrial process. Since most industrial applications pursue lucrative objectives, a strong constraint exists on how contributors are rewarded. When studying the crowdsourcing landscape two payment methods can be identified.

- *Best-gets-paid system*: Usually, in this type of system, only the best workers are rewarded. In general, the system provides the tools to present ideas or solutions to a specific problem and a voting system for the crowd to decide what the best proposals are. This philosophy usually allows for reducing costs drastically and obtaining very good quality. Nevertheless, the approach is unfair to most workers, for only one contributor gets paid.

- *Pay-per-work system*: In this case, workers are rewarded for the amount of work done. An example is Amazon's Mechanical Turk, where workers execute Human Intelligence Tasks (HITs) and get a predefined amount of money based on the completion of the task. The downside is that this approach does not reward the quality of the work (Mason & Watts, 2009; Rogstadius et. al., 2011).

These systems both have their flaws, either with ethical issues or the lack of incentives to deliver high quality work. What we propose is a *pay-per-quality system*, where payout is based on a combination of the *pay-per-work system* and the quality associated with it. The problem with rewarding quality lies in the measurement of the quality. In our rewarding mechanism we link the trustworthiness ranking of users with the ranking index of the worker. This leads to a fair rewarding system, which introduces the possibility to grow as a translator in such a system, and be recognized and rewarded for delivering higher quality work.

4.4.1 Cost of translation

The current industry average cost per word lies at around €0.09 per word, and can rise up to €0.15 when work has to be outsourced to an external party. The costs for translating a 150 page manual consisting of 75.000 words is €6.750,- when using in house translators, or €11.250,- when outsourcing the work.

To calculate the maximum cost per word for the crowdsourcing platform we take into account two things. First the cost for the crowdsourcer, and second a competitive pay per hour for crowdworkers.

The payout for each different user differs based on the type of work they do and rank they have (figure 12). A Post Editor with a QI rank of 90 or higher will receive €0,02 cents per word. While a Post Editor with a rank between 70 and 80 will receive €0,0125 cent per word. Using this new approach, by applying two AVI-units each with 3 verifiers, the cost of

translating a 150 page manual consisting of 75.000 words will be stated at €3.632.81,-, which is a reduction of nearly 50%. As post-editors translate on average 450 words per hour (CommonSenseAdvisory, 2012) the payout reaches an average of €9.00,- per hour. Which is considerably higher than the average pay of €1.44,- crowdworkers on Mechanical Turk make (Ross et. al., 2010).

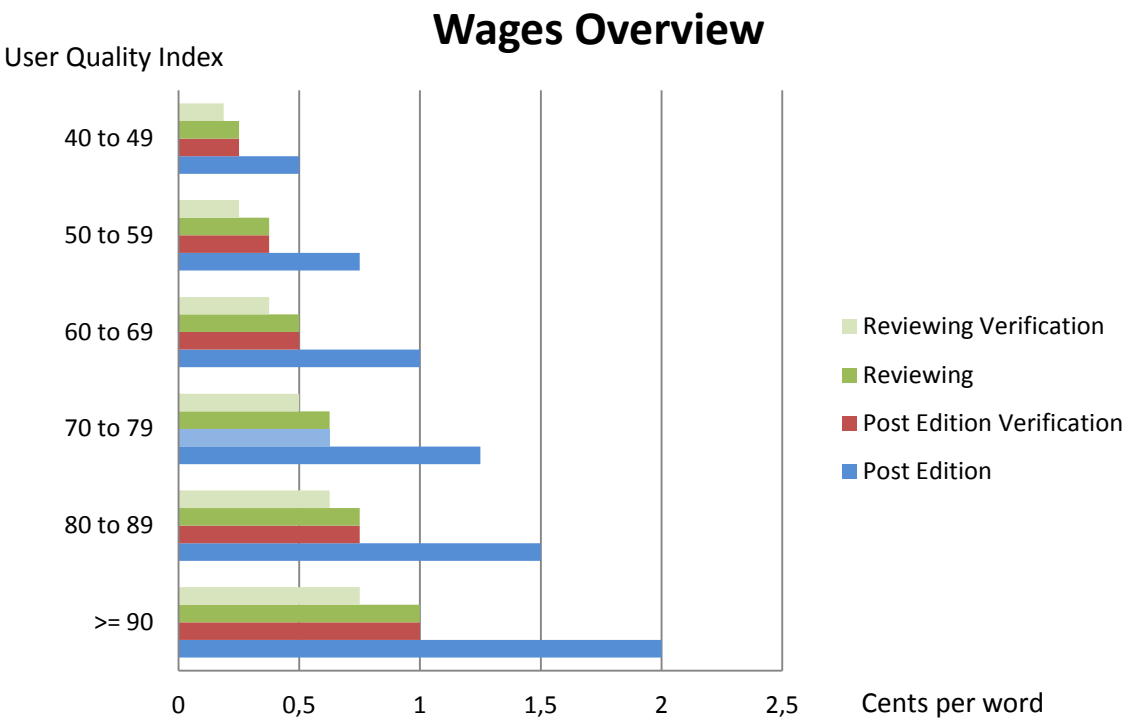


Figure 12: Costs in cent per word, set off against the different user ranks

4.5 Prototype

The design is built into a working prototype to test the functioning of the system. The architecture of the system revolves around a number of key concepts, namely the *task management*, *user management* and *ranking management*, a rewarding module linked to users and ranking, a task manager responsible for document and task management. The task manager is linked to unit managers who control the different phases in the system. The complete system architecture and enterprise relationship diagram can be found in appendix A and B.

The implementation of the system follows the by Reenskaug (1979) introduced Model View Controller (MVC) architecture (Figure 13 on the next page). The MVC is popular for its clear distinction between the representation of the knowledge and information in the system (the

model), the graphical representation of tasks and information through the front-end (the view) and the control mechanism of managing data flows between user and system (controller). The prototype system is built on Java technology and uses J2EE Javabeans, the encapsulation of data objects in the database to represent the model, Servlets to represent the controller and JSP pages to form the view.

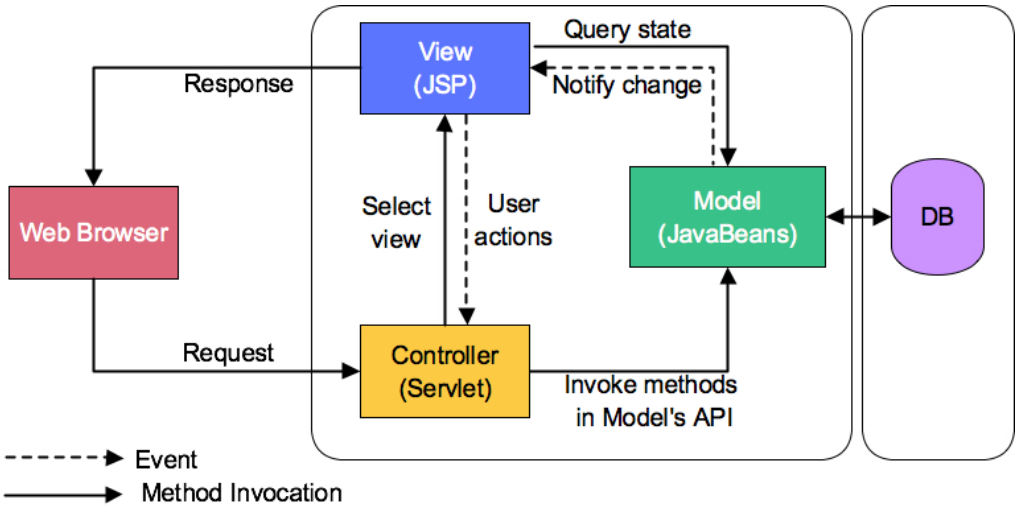


Figure 13: MVC architecture

To give a preview of the platform we provide a number of screenshots, please note the system is designed for Catalan and Spanish users and is translated to their native tongue. The web-applications design is depicted in Figure 14, which shows the home page of the website. Here the user can register to the system, or login to review tasks that have not yet been completed. The post edition task is displayed in figure 15. The user is presented with the machine translated text one the left side, and the English text on the right side. After the work is submitted it is forwarded to a verifier (figure 16). The verifier can check the different types of errors he finds in statements and provide feedback. In the final phase (figure 17) the Post Editor or Reviewer will review the definite erroneous material and improve the work. On this screen he will receive an overview of the errors that verifiers selected and can read comments about that specific segment.

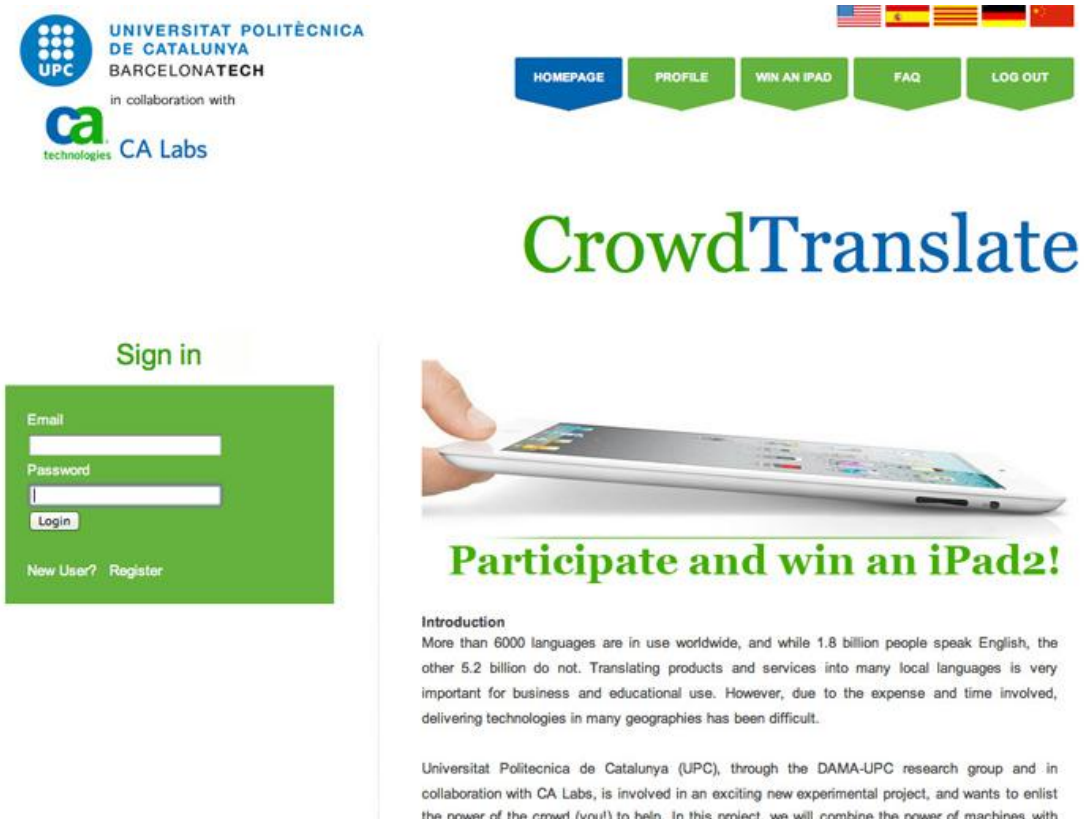


Figure 14: Website interface

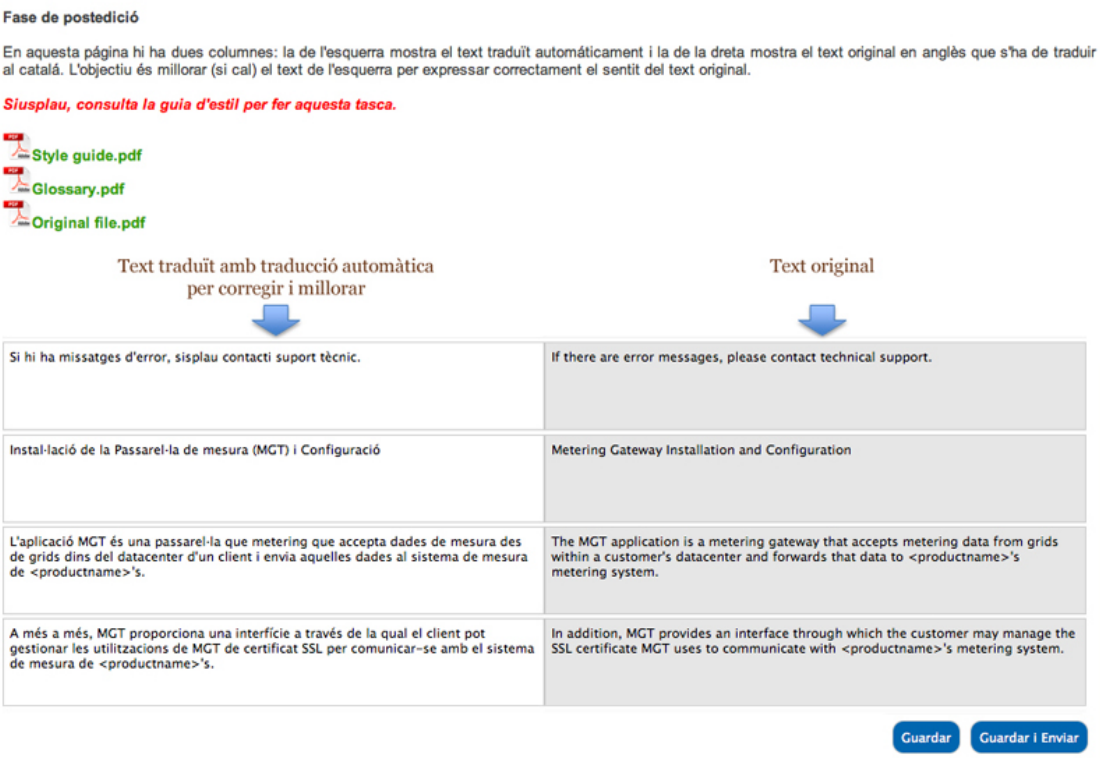


Figure 15: Post Editor and Reviewer Interface

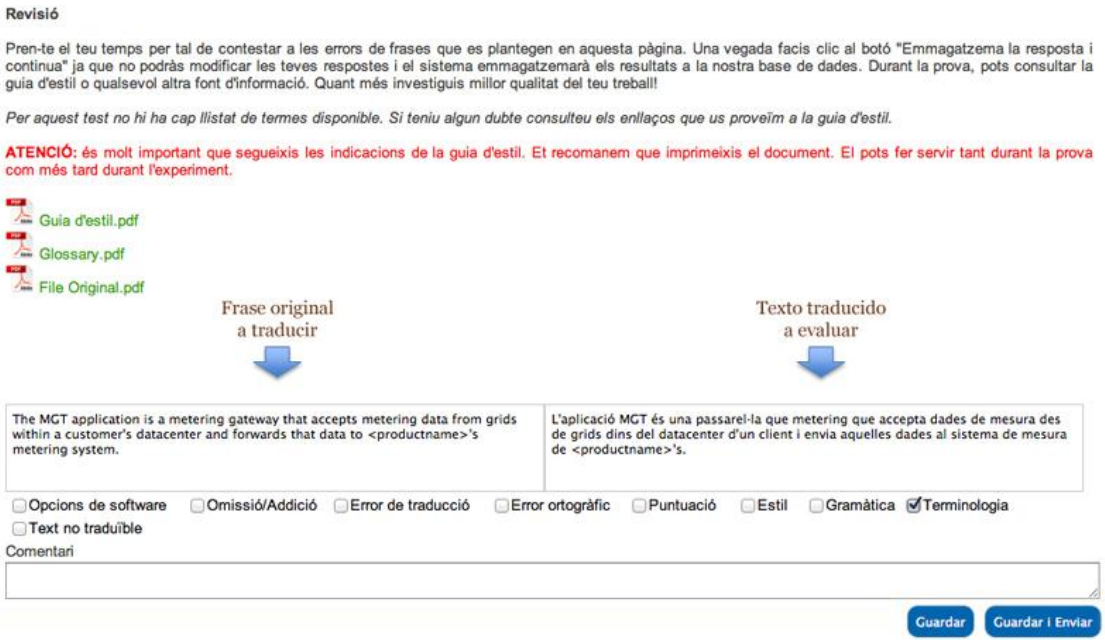


Figure 16: Verification Interface

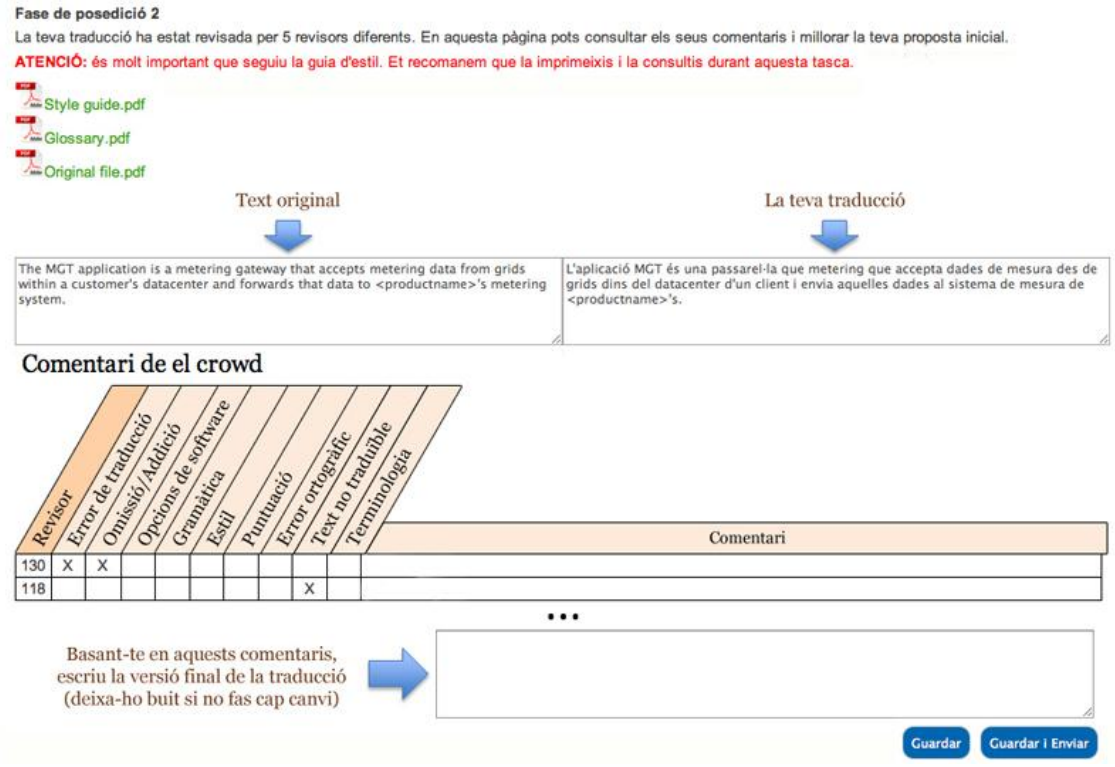


Figure 17: Post Editor and Reviewing Improvement Interface

5 Platform and AVI-unit evaluation

5.1 Introduction

In chapter four, we introduced a *crowd translation* platform to facilitate industrial localization using crowdsourcing. This system is based on the newly introduced AVI-units to provide better quality assurance and improve the work crowdworkers deliver. In this chapter we set out to get an indication of the workings of this crowd translation platform and more specifically, by evaluating the workings of the AVI-unit. To gather enough information to have a statistical base for our findings, we scope this experiment to researching the workings of a *single* AVI-unit. We do this by measuring three different segments of the AVI-unit, as indicated in figure 18. We perform our 1st quality measurements[11] after the a user has performed an initial translation. After this step, we measure the performance of the verifiers, and determine how the verification unit performs. Lastly, we see what impact the verification and improvement phases have had on the translation, and perform a 2nd quality measurement after the improvement phase.

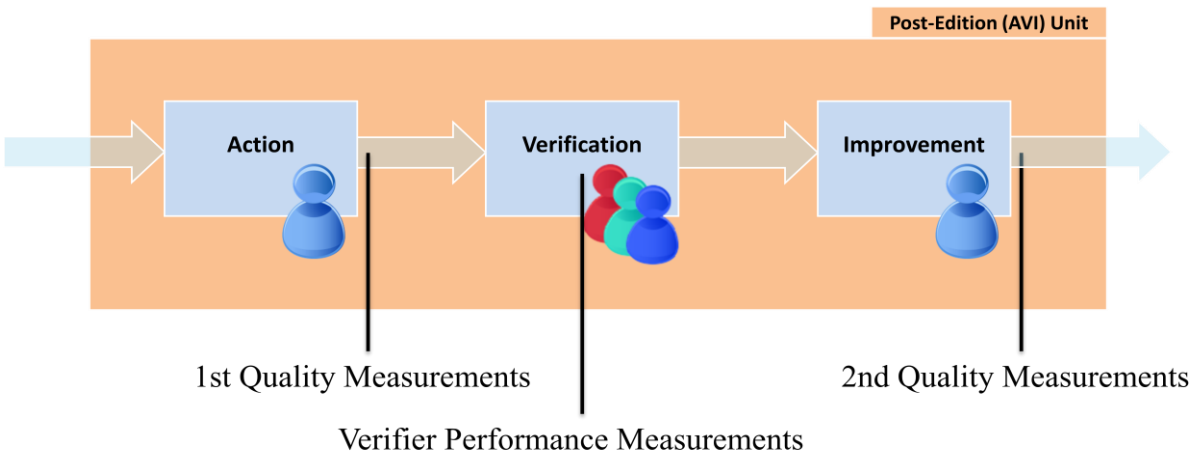


Figure 18: Points of measurement to evaluate the working of the AVI-unit

For our experiment we setup two different language experiments and verify a number of hypotheses. The experiments consist of translating a technical manual from English to Catalan and from English to Spanish, both using an identical experimental setup. In this chapter we address the different hypotheses, the relevant variables, the experiment design we used and the method of selecting and assigning users in the experiment.

[11] The quality measurements are performed by professional translators from CA technologies, using the evaluation toolset provided in appendix C, D and E.

5.2 Hypotheses

We identify three different kinds of hypotheses. First the hypothesis regarding the dataset and the difference on quality we might find if we work with crowdworkers with a different native tongue. Second, two hypotheses regarding the performance of the AVI-unit. What is the quality we find, and does the AVI-unit improve the translation quality? Lastly, we identify two hypotheses regarding the performance of the verifiers in the AVI-unit. How well do they operate when compared to professionals, and what effect do the different configurations of users in this phase have on the final quality? Each of the variables in this section are discussed in chapter 5.3: Variables.

5.2.1 Hypothesis regarding the dataset and differences between languages

As we perform the same experiment twice, with different languages, we first determine whether the number of errors for both differ significantly or not. This is important for a number of reasons. First, we want to know whether we can use both experiments as one coherent dataset in the analysis phase. If we find that both datasets are equal this means more options in terms of statistical analysis are available. Testing this is important, as it is not unlikely that the quality of translations for different languages will differ. Because, since we are dealing with crowdworkers and not professional educated translators, the quality of the translation depends highly on the knowledge the typical crowdworker has of both source and target language. As a European Commission (2005) report on 'Europeans and Languages' states, the knowledge of both the native tongue and the English language differs greatly within countries. For example, about 85% of the adults in the Nordic countries, such as Sweden, Denmark and the Netherlands claim to speak English, while in countries such as France this is 34% and in Spain more around 20%. Second, languages can differ greatly in the way they describe the world or how cultural subjects are addressed (House, 2001). For example when certain words in another language can have a second, different meaning, perhaps unknown to a less experienced translator. However, in our experiment we are translating into Catalan and Spanish, languages closely related and spoken by the people from the same country (Spain). Our hypothesis on the difference of quality from crowdworkers in translating to these languages is therefore:

> $Hypothesis_1$ - The total error points created in the Post Edition AVI-unit by Spanish and Catalan crowdworkers are similar

5.2.2 Hypotheses regarding the performance of the AVI-unit

To evaluate whether an AVI-unit improves the quality of translations we performed an experiment in our online crowd translation platform that implements a single iteration of an AVI-unit (the simplest pattern as indicated in figure 18). We measure the quality of the translation by calculating the Total Erorr Points (TEP), a weighted score that takes into account the severity of the error. To determine whether the translation has improved we observe whether the TEP decreases significantly.

> Hypothesis$_2$ - The use of the Post Edition AVI-unit significantly decreases the number of Total Error Points in a translation.

Normally, to provide a verbal quality scale (a scale stating whether a text is a 'Reject' up to of 'Excellent' quality) for a translated text, the TQI is calculated (Schiaffino & Zearo, 2006). The system for using TQI's is of added value as it helps when comparing texts of different text sizes. This, because The TQI formula presents a numerical value that is based on the TEPs of a text while taking into account the *size* of the text. However, in our experiment we are dealing with a text of exactly the same number of words, thus the use of the TQI would involve performing additional calculations without providing extra information as opposed to only analyzing the TEPs and basing the verbal quality scale on the errors.

To use the TEPs as a measurement scale we need to calculate what number of errors couples with the verbal quality scale. In the software industry this value is set where a text can contain at maximum 1.5% errors for being a text that is still improvable to an acceptable standard (CA Technologies, 2012). Thus for a translation to be borderline acceptable the number of weighted errors points for a 350 word text should preferably be no more than 1.5% of the text, or equivalent to 5,3 TEPs. The quality scale for a 350 word text, and their associated error points is depicted in figure 19.

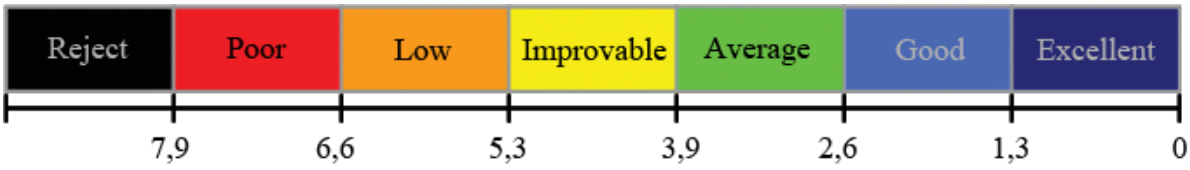


Figure 19: Total Error Points and the respective verbal quality scale (between 'Reject' and 'Excellent') for a 350 word text (Schiaffino & Zearo, 2006).

Unfortunately, we only know the current industry standard for professional translations, after completing *both* the Post Edition and Reviewing phase, when they score between 1.3 and 0 TEPs for a 350 word text (thus in verbal terms resembling an 'Excellent' translation). In our experiment, since we only tests the working of the AVI-unit for the *Post Edition phase* we measure whether the AVI-unit delivers translations of a improvable standard or better. This implies the translation is improvable, for example by adding the reviewing AVI-unit. Our third hypothesis is therefore.

> Hypothesis$_3$ - The use of the Post Edition AVI-unit for a 350 word text delivers translations with Total Error Points of 5.3 or less.

5.2.3 Hypotheses measuring the performance of the verifiers in the AVI-unit

The verification phase of the AVI-unit uses three verifiers to find and identify mistakes in the translated text of the Post Edition crowdworker. To diagnose the performance of the AVI-unit, we study how well a verifier mimic's the behavior of a professional reviewer. To obtain such a characterization it is useful to think of the verifiers performance in terms of an observation that needs to be compared to a dataset where the correct answers are known. Thus the evaluations of the professional reviewers are the ground truth and we explore how well our verifiers perform against this ground truth. To analyze these kind of systems we can perform a series of binary classification[12] tests (Shawe-Taylor & Cristianini, 2004). These are statistical analysis tests widely used in the field of medicine experiments[13] and consist of calculating the Precision-, Recall-, Specificity- and Accuracy classification metrics.

> Hypothesis$_4$ - Higher ranked verification users perform better on the binary classification metrics than lower ranked users.

To further determine the working of the verification phase we find out what impact the different group compositions have on the overall improvement of the text. We do this by proposing two new and simple formula's which calculate a *fixing rate* and a *destruction rate*. Two metrics to show what proportion of the errors were fixed after the verification phase and what proportion of errors were newly introduced.

[12] Binary classification is the act of classifying the members of a give dataset two groups on the basis of whether they have a property or not.
[13] An example is to see how accurate or precise a medical test truly identifies whether a person was ill with the diagnosed disease or not.

Hypothesis$_5$ - Higher ranked verifier groups perform better on the fixing- and destruction rate metrics than lower ranked groups.

5.3 Experiment Design

For the experiment design, we use the definition Campbell and Stanley (1963) introduced in their book on Experimental and Quasi-Experimental design for research. They state that an experiment is "*a portion of research in which variables are manipulated and their effects upon other variables are observed*". The design of the experiment is based on Blumberg's Business Research Methods (Blumberg, Cooper, & Schindler, 2008) and Campbell's and Stanley's Experimental and Quasi-Experimental design for research (Campbell & Stanley, 1963). Blumberg proposes a method where key activities need to be addressed for results not be questioned on validity and reduce the main disadvantages of experiment research. The following sections discuss the research variables, the experiment design and the selection, assignment and segmentation of users.

5.3.1 Variables

For the research variables we describe independent (IV) and dependent (DV) variables. By altering the independent variables the result on a dependent variable can be measured. In the experiment we measure the effectiveness of the AVI-unit as a *translation method* on the number of errors and quality of a text. In addition, we test the workings of the verification phase in the AVI-unit. The different types of users from the crowd participating in our system will be performing translations. To address blocking[14] the crowdworkers have been divided into three groups based on their translation capabilities. Each has been assigned the *rank* of A (high), B (medium) or C (low). This is done to reduce the confounding effect that a person's ability to translate will have on the results. The variables for this research have been visualized in the following conceptual models (figure 20 to 22) and each variable and its respective level of measurement is addressed in the section below.

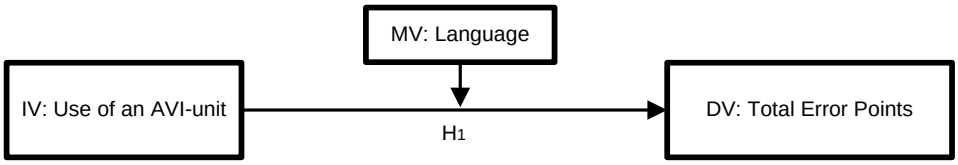


Figure 20: Conceptual model for hypothesis 1

[14] Blocking in statistics is the arrangement of experimental units into blocks.

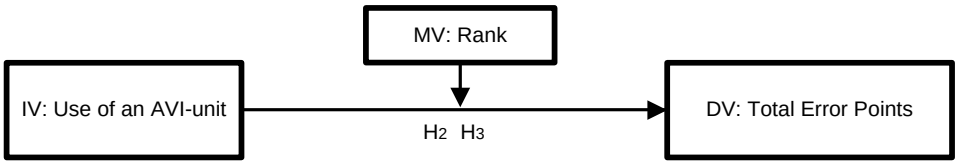


Figure 21: Conceptual model for hypotheses 2 and 3

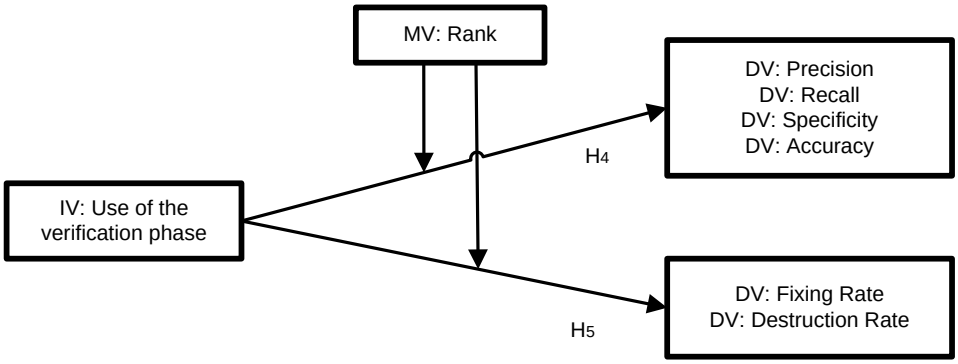


Figure 22: Conceptual model for hypotheses 4 and 5

The variables and their measurements are depicted in table 6.

Table 6: Levels of measurement

Variable	Type	Measurement	Value
Use of an AVI-unit	Independent	Nominal	Yes (1); No (0)
Use of the verification phase	Independent	Nominal	Yes (1); No (0)
Rank	Moderating	Ordinal	(A); (B); (C)
Language	Moderating	Nominal	(Catalan); (Spanish)
Total Error Points	Dependent	Interval	(0,00 to ∞)
Fixing Rate	Dependent	Ratio	(0,00 to 1,00)
Destruction Rate	Dependent	Ratio	(0,00 to 1,00)
Binary Classification Metrics			
Recall	Dependent	Ratio	(0,00 to 1,00)
Precision	Dependent	Ratio	(0,00 to 1,00)
Sensitivity	Dependent	Ratio	(0,00 to 1,00)
Specificity	Dependent	Ratio	(0,00 to 1,00)

Use of an AVI-unit (IV)

A nominal variable indicating whether the final translation was created using the AVI-unit, thus adding the Verification and Improvement phase, instead of just performing the Post Edition (Action) phase.

Use of the verification phase (IV)

The use of the verification phase is a nominal variable, where a crowdworker uses the mechanics of identifying errors, selecting them and commenting on how to improve them.

Rank (MV)

A user's rank is determined through a *translation capability test*. This test was mandatory for every participant in the experiment and would determine the rank the user had. In the test we provided them with a style guide and a glossary to familiarize themselves with the translation terms and rules. Each user was tested on their ability to find the best translations, find grammatical errors and identify style issues in a text. The test is based on the evaluation procedure in the professional translation department at a large software company. To reduce the time required for making the test we created a smaller test consisting of 26 questions. The test contains three segments consisting of translation and verification questions. The test was translated and reviewed by professional translators so both Catalan and Spanish users take the exact same test. More details on the *translation capability test* can be found in appendix J.

Language (MV)

The language variable is of a nominal level of measurement and indicates what language the *native tongue* of a crowdworker is.

Total Error Points (DV)

Calculating the TEPs in a text is not as straightforward as adding points for every error that is made. For a more comprehensive and precise judgment of the text the equation for the TEPs takes into account the severity of the error as well. The formula for this calculation is depicted in equation 1. In total the system identifies 46 different error types from 7 different error categories (appendix D). For each error that is found one of 4 different *severity percentages* has to be chosen (appendix E). Thus a category 1 error has a severity percentage of 60%, a

category 2 error of 20%, category 3 of 12% and a category 4 error of 8%. Below an example is given on how this is calculated.

Equation 1: Calculating the Total Error Points.

$$\text{Total Error Points} = \sum_{Error\_Types}^{7} \left(10 \text{ x } \sum_{Severity\_Perc.}^{4} (Severity\_Percentage \text{ x } Error) \right)$$

Example:

In a text the following errors are found, two style errors of category 4 (thus 8% severity), a *grammar* error at category 2 (thus 20% severity) and a *software options* error of category 3 (thus 12% severity). The TEPs in this text are as follows:

Total Error Points = round(10 x (8% x 2)) + round(10 x (20% x 1)) + round(10 x (12% x 1)) = 1.6 + 2 + 1.2 = 4.8

Fixing rate

The fixing rate we propose is the proportion of sentences that contained an error in the post edition phase and were successfully corrected in the post edition improvement phase. It is an indication on how successful the Verifiers and Post Editor Improvers collaborate. We calculate the fixing rate according to the following equation (equation 2).

Equation 2: Calculating the Fixing efficiency

Fixing Rate = Errors found in text / Errors fixed after verification

Destruction rate

The destruction rate we propose is the proportion of sentences that were correct in the Post Edition phase and saw new errors introduced after the Post Edition improvement phase. It is an indication on what can happen when multiple verifiers indicate an error and how the Post Editor improver reacts to this. We calculate the destruction rate according to the following equation (equation 3).

Equation 3: Calculating the Fixing efficiency

Destruction Rate = Errors introduced / Total sentences without an error

Binary Classification Metrics (DV)

To calculate binary classification metrics we first need information about the actual and predicted errors by crowdworkers in the verification phase. To map this information we use a table of confusion (Kohavi and Provost, 1998) as illustrated in figure 23. Within the table four different factors are described:

- TP is the number of **correct** predictions that a sentence is correct (positive),
- FN is the number of **incorrect** of predictions that a sentence is wrong (negative),
- FP is the number of **incorrect** predictions that a sentence is correct (positive), and
- TN is the number of **correct** predictions that a sentence is wrong (negative).

		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Figure 23: A table of confusion

The binary classification metrics provide information on the success the crowdworker has on finding the right errors. Here, Precision (equation 4) is the probability that if the verifier says that the sentence contains an error, it actually contains one. Recall (equation 5) is the proportion of all sentences containing errors (as indicated by a professional translator) that were actually identified by the crowdworker. Specificity (equation 6) is the proportion of correct sentences identified as such, Finally, and most importantly, the accuracy (equation 7) is the proportion of the total number of predictions that were identified correctly by the crowdworker.

Equation 4: Calculating the Precision metric

$$Precision = TP / (TP + FP)$$

Equation 5: Calculating the Recall metric

$$Recall = TP / (TP + FN)$$

Equation 6: Calculating the Specificity metric

$$Specificity = TN / (TN + FP)$$

Equation 7: Calculating the Verifier Success Rate

$$Accuracy = (TP + TN) / (TP + FP + FN + TN)$$

5.3.2 Experiment Design

Our experiment design follows the three phases of the AVI-unit and is depicted in figure 24. For the experiment we assign the ranks A, B or C to the users as explained in section 5.3.1. After completing the categorization, we assign the workers to the two phases of the translation process. The experiment knows two moments where professional translators checked the current quality of the translation by calculating the TEPs. The first being after the Action Phase, or *before the manipulation* and the second being after the Improvement Phase, or *after the manipulation* (figure 23 on the next page). As discussed in section 5.2 we perform this experiment twice, both for the Catalan and the Spanish language. For each experiment 12 Post Editors, 36 verifiers and 12 Post Editor improvers have participated. Totaling 24 Post Editors, 72 verifiers and 24 Post Editor improvers. In total 24 texts will be translated, containing 648 sentences and 8400 words. A further description of the experiment design of figure 24 is added in table 7. Which holds a more detailed overview of the steps that crowdworkers undertake in each of the three phases.

Post Edition						Verification		Post Edition Improvement					
User	Rank	Text	Sentences	Words	Evaluate quality of work by professionals	Type		User	Rank	Text	Sentences	Words	Evaluate quality of work by professionals
1	A	1	27	350	->	ABC	->	13	A	1	27	350	
2		1	27	350	->	AAA	->	14		1	27	350	
3		1	27	350	->	BBB	->	15		1	27	350	
4		1	27	350	->	CCC	->	16		1	27	350	
5	B	1	27	350	->	ABC	->	17	B	1	27	350	
6		1	27	350	->	AAA	->	18		1	27	350	
7		1	27	350	->	CCC	->	19		1	27	350	
8		1	27	350	->	BBB	->	20		1	27	350	
9	C	1	27	350	->	AAA	->	21	C	1	27	350	
10		1	27	350	->	BBB	->	22		1	27	350	
11		1	27	350	->	CCC	->	23		1	27	350	
12		1	27	350	->	ABC	->	24		1	27	350	
Totals		12	324	4200		36 users		Totals		12	324	4200	

Figure 24: Experiment design using crowdworkers with three different quality ranks (A,B and C). The experiment design is identical for both the Catalan and the Spanish experiment.

Table 7: Phase description for a single language experiment

Post Edition (Action)	In the post editing phase 12 users will translate a one page technical manual[15] which can be found in appendix K. The users are selected based on their ranks, respectively 4 A, B and C users. Each text consists of 27 sentences and 350 words.
Post Edition (Verification)	In the post editing verification phase the work of the 12 texts that have been translated in the Post Edition phase will be verified. In total each text will be verified by 3 people, who indicate what sentences of the translation contain errors and other mistakes. For each text a different combination of verifiers is selected. The combinations are based on the ranks of users and are: AAA, BBB, CCC or ABC. This can be used to determine the impact of rank has on the quality of the translation. After the verification is completed, the sentences with errors are sent to the Post Editor improver.
Post Edition (Improvement)	The Post Editor improvers are presented with the feedback from the verifiers. They can use this feedback to improve the work and submit a higher quality translation. Post Editors and Post Editor improvers always scored the same score on the translation test determining their rank. **Note:** *To reduce the workload of the Post Editors the post edition improvement are different individuals. To guard the experiment data these people have been selected to have the same rank as the original Post Editor of that specific text.*

[15] The manual was designed by professional translators to present an average page in a technical manual.

5.3.3 Selecting, Assigning and User Segmentation

To be able to control the external validity of the experiment we apply random sampling techniques. Due to the dataset we have, normal random sampling is not the best option and could threaten the validity of the results. For example, suppose we have 40 A ranked users with quality ranks between 60 and 95. If we need only a relatively small sample out of these 40 users, for example 5, we could jeopardize the spread of users in our sample. We could end up with only high ranking users (5 users, each ranked between 90 and 95), or only low ranking users (5 users, each ranked between 60 and 65). As there is a difference in quality between 60 and 95 we prefer having a good spread of the 5 users we select for our sample. Systematic sampling addresses this problem through the following formula, where every k^{th} element is selected out of a sample size *n* and a population size N.

$$k = \frac{N}{n}$$

Example

We use the example of 40 users with different quality ranks between 60 and 95. The list of users is ordered by rank. If we require a sample of 5 out of our population of 40 we select every 40/5 = 8^{th} element. In our sampling frame the starting point is chosen at random between 0 and 8. After that every 8^{th} user is selected. With 6 as a starting point, the users 6, 14, 22, 30 and 38 are selected.

This same approach is used when contacting potential translators and verifiers for our experiment. In total the number of people who participated in the initial test exceeded the number of people we required in the experiment. If a selected user did not reply or did not finish the task the closest ranking user was selected to replace him.

6 Results

6.1 Introduction

Important aspect of this experiment is the use of differently *ranked* users. Through our hypotheses we determine how differently ranked users operate in our system and what configurations of users perform best. To determine these ranks we had every single participant for the experiment perform the *translation capability test*. In the first section we discuss the results of the capability test and how we assigned them the ranks of A, B and C.

The second section of the results chapter starts by discussing population demographics, data quality and outliers. After which the experiment results in which the hypotheses form the order of discussion.

6.2 Determining the crowdsourcers quality ranks

Before crowdworkers could participate they had to do the *translation capability test*. Figure 25 shows the histogram and normal distribution line for the Catalan and Spanish translation test, where for the Catalan test 100 users completed the test and for the Spanish version 114 users completed the test. If a user scored 1, he had all questions on the test correct.

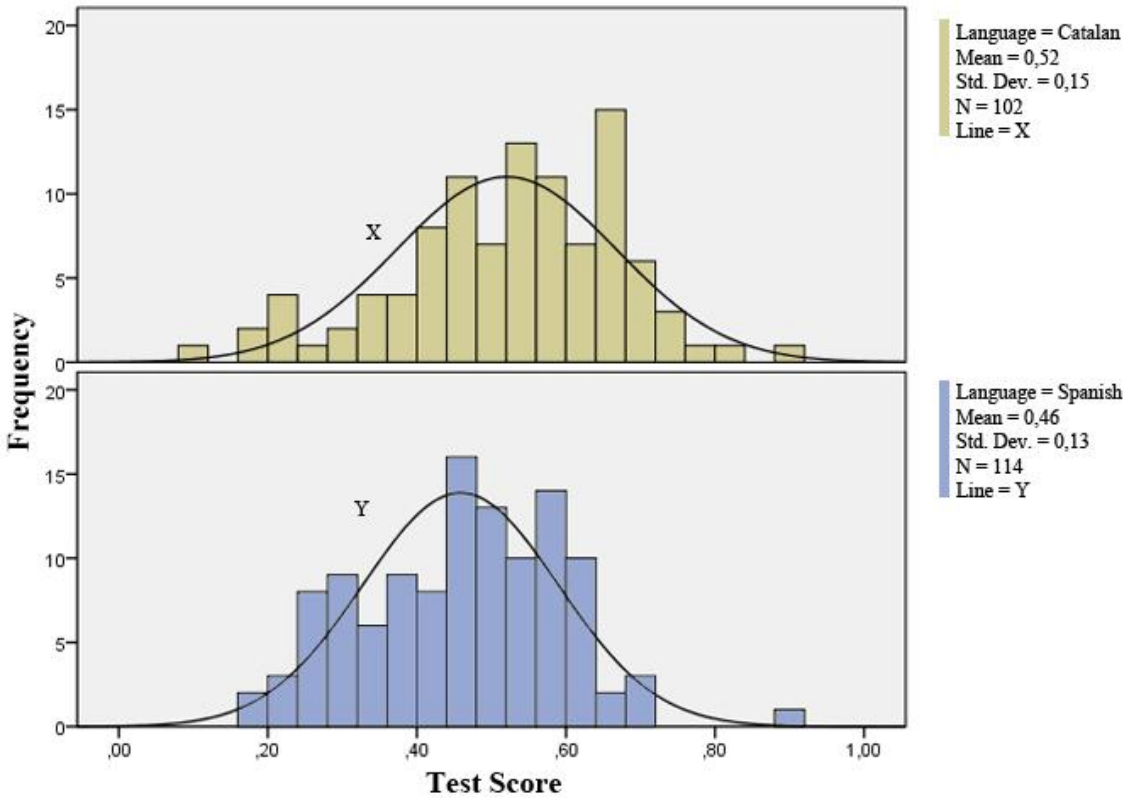


Figure 25: Histogram of the normalized translation test scores

The detailed test results are depicted in table 8 and 9, and show the normalized scores for the first-, second-, third section (appendix J) and combined score of the test. The relative high amount of correct questions on the first section (0,68 for Spanish and 0,79 for Catalan) can be explained through its multiple choice nature. The second and third section show very similar results suggesting no anomalies were present in the test. Though it shows people had the most difficulty in the second part of the test, which tests the ability to identify specific errors in a translation. Interesting is the discrepancy of the first and second section and the combined scores between the Catalan and Spanish test.

Table 8: Normalized test results for the Spanish language test

	Section 1	Section 2	Section 3	Combined
Mean	0,68	0,34	0,44	0,46
Median	0,71	0,33	0,44	0,47
Mode	1,00	0,42	0,56	0,58
Sdev	0,27	0,12	0,18	0,13
N	114	114	114	114

Table 9: Normalized test results for the Catalan language test

	Section 1	Section 2	Section 3	Combined
Mean	0,79	0,40	0,43	0,52
Median	0,86	0,40	0,44	0,53
Mode	1,00	0,47	0,44	0,47
Sdev	0,24	0,15	0,17	0,15
N	102	102	102	102

To test whether the difference is significant we first perform the Shapiro-Wilk (Shapiro & Wilk, 1965) test for normality, as dealing with a normal distribution is the criterion for being able to perform a t-test. The Shapiro-Wilk test is considered to be the standard test when dealing with datasets with an N below 2000. Based on our Shapiro-Wilk test we conclude that both the results for the Catalan ($p = 0,109$) and the Spanish ($p = 0,073$) test will have their hypothesis being rejected, concluding that they are normally distributed (appendix L). To test the significance we performed an independent two-samples t-test (Snedecor & Cochran, 1989) with equal variances assumed. The results can be seen in table 10, with the full printout in appendix L.

Table 10: Independent two-samples t-test for the different translation test sections

	t-value	p-value	DF	N
Section 1	3,433	0,001	214	216
Section 2	3,524	0,001	214	216
Section 3	-0,129	0,898	214	216
Combined	3,059	0,003	214	216

The first observation we can make is that the difference between section 3 is not significantly different (p = 0,898). This is true because the t-value does not exceed the critical t value of 1.9673 (Snedecor & Cochran, 1989). A reason for the similarity in section 3 could be behind the nature of these questions, as section 3 tested *reviewing* capabilities, and not *translating* capabilities. In addition, as users are only dealing with their native language and had to pick between only 4 possible errors to identify, it was much simpler than the other questions, possibly explaining the similarity in scores.

The second observation we make is that the results for the other sections are significantly different. Where for each section and the combined set the t-values exceed the critical t value of 1.9673. This means that Catalan users scored significantly better than Spanish users on the capabilities test. The explanation for this difference is most likely found in the expertise of the crowdworkers (table 11). As within the Spanish crowdworkers only 11% of the users were working in the translation industry, while for the Catalan experiment this was close to 0,18%. In addition we can see that the number of amateurs participating in the Spanish experiment are slightly higher with 60% as opposed to 58% in the Catalan experiment. In addition, the percentage of the students in the Spanish was slightly higher, which might have skewed the results to a lower ranked. Especially since students participating in the Spanish experiment were recruited at a translation department of the University of Barcelona and were requested to perform the translation test within a single hour of class, possibly pressuring them to hurry up and paying less attention to details.

Table 11: Proportion of crowdworkers and their expertise related to linguistics

	Spanish	Catalan
Professionals	0,11	0,18
Students	0,29	0,26
Amateurs	0,60	0,58

This difference in the *translation capability test* results could endanger the comparability of both experiments. Especially as the Catalan experiment had already completed when the Spanish tests were being done and the Catalan users had already been put in the respective groups of A, B and C based on the number of Standard Deviations from the mean (Std deviation. 0,5 and higher above the mean for A ranked users, Std. deviation. 0,5 and lower below the mean for rank C and everything in between for rank B). Using this same approach to group the users in the Spanish experiment would result in the groups being of different quality, which means comparing results from the Spanish experiment to the Catalan experiment would be difficulty.

To address this problem the Spanish users have been put into the groups based on the same thresholds that were used for the Catalan users. The number of users and their respective ranks are shown in Table 12. This approach ensured that the groups of users both within the Catalan and Spanish experiment were of the same rank, making the experiments better comparable.

Table 12: Number of users and their respective ranks

Rank	Std. deviation from the mean	Score range on 0-100 scale	Catalan Users	Spanish Users
A	0.5 and more above	0,58 and above	38	26
B	Between 0.5 below and above	Between 45-57	34	44
C	0.5 and less below	0,44 and below	30	44
		Totals	102	114

6.3 Experiment results

6.3.1 Crowdsourcer demographics

Divided over the Catalan and Spanish experiment, in total 24 users (8 of the rank A, B and C) performed a translation on a text containing 27 different sentences and 350 words. We used 72 users to verify this work, after which 24 more people were used to improve these texts (Note, as discussed in the experiment design, these 24 are different, but similarly ranked, users). The participants in the experiment were approached using several formal and informal networks. Examples are: a post on the CA Intranet, LinkedIn interest groups, Facebook events, university mailings, recruiting through Mechanical Turk and mouth to mouth advertising. The average age of the crowdworkers was 30 years old, with the oldest being 67 years old and the youngest 19 years old. Table 13 gives an overview of a number of characteristics of the crowdworkers.

The discrepancy between profession and experience is that experience only deals with students studying to work in the field of translation or linguistics, while the profession also includes other students and researchers. Although it was possible to indicate to have a low understanding, none of these users, who also completed the skill test, were randomly selected to participate in the experiment. On average Post Editors spent 58 minutes on their task, while Post Editor improvers took 38 minutes. Combined, the users spent over 400 hours testing, translating, verifying and improving texts on our platform.

Table 13: Characteristics of crowdworkers

Characteristic	Value	Percentage
Knowledge of the English language	High	72%
	Medium	28%
	Low	0%
Experience regarding the field of translation	Professional translator or similar	14%
	Amateur	61%
	Student	25%
Profession	Related to IT	23%
	Related to University studies	30%
	Related to IT and Translation	19%

	Related to field of Translation	17%
	Other	11%

Test for Normal Distribution

Before being able to perform statistical tests on our data a test for normality is required. Based on the Shapiro-Wilk (table 14) test we conclude that all the data sets will see the Shaprio-Wilk hypothesis be rejected, concluding they are all normally distributed (appendix L).

Table 14: Test for Normality using Shapiro-Wilk (Shapiro & Wilk, 1965)

	Shapiro-Wilk PE	DF	N	Shapiro-Wilk PEI	DF	N
Catalan	P = 0,328	12	12	P = 0,331	12	12
Spanish	P = 0,219	12	12	P = 0,233	12	12

Test for Outliers

To identify outliers we perform the Grubbs test for outliers (Grubbs, 1969). The test can be performed on a normally distributed dataset and essentially detects which variables exceeds the z-value of 2.032, or is below the z-value of -2.032. Based on the analysis on the PE and PEI phases for both the Catalan and Spanish users we can identify one outlier (Table 15).

Table 15: Outliers according to Grubbs' test for outliers (Grubbs, 1969)

User id	Language	Rank	Phase	Grubbs' (z) value	p-value
126	Spanish	B	Post Edition Improvement	2,233	0,05

This user can be characterized by creating a very high number of errors, some of which with the (rare) highest severity levels. In total the user created 44,4 error points with sometimes up to seven different errors in a sentence. Fascinating fact is that the user had scored a rank of B on the translation test, meaning he or she either cheated on the test or did not approach the task with the same attention as on the translation test. A factor which strengthens this hypothesis is the time spent on the task. While the average of Post Edition Improvers was to spend 38 minutes improving texts, this user did it in only 8 minutes, and ended up creating more errors in the text as there were before. As a result of these observations we removed this outlier from the dataset, ending up with 12 Catalan and 11 Spanish data points for the Post Edition Improvement phase

6.3.2 The impact of languages on translation quality

To determine what impact language has on translation quality, we propose testing the difference between Catalan and Spanish quality.

H_1	The total error points created in the Post Edition AVI-unit by Spanish and Catalan crowdworkers are similar

As we already established the datasets are normally distributed in the previous section, we now continue to measure the equality between the Spanish and Catalan experiment. To test this we performed an independent two-sample t-test (appendix L) to determine whether the means of the TEPs created by Spanish and Catalan Post Editors and Post Editor improvers differed significantly or not. For the test equal variances are assumed and the test results can be found in table 16.

Table 16: Results two sampled t-test for equality of means between Spanish and Catalan users

Catalan Mean TEPs	Spanish Mean TEPs	Phase	DF	t-value	p-value	N
18,47	22,18	Post Edition	21	-0,913	0,754	23
14,27	15,34	Post Edition Improvement	21	-0,317	0,372	23

As for both the t-value's lie below the critical t-value of 1.9673 with a p-values of 0,283 and 0,699 we can state that the means for total error points between Spanish and Catalan users are not significantly different and our hypothesis is not rejected. One can argue the difference in means between the Catalan and Spanish Post Editors is an indication of a difference. However, this is caused by a 'near' outlier, delivering an almost perfect text in the Post Edition phase (appendix L, Catalan user 8). Concluding, these result were to be expected. Since the users have been selected to participate based on an identical translation test, and were subject to the same criteria. In addition, Catalan and Spanish are related languages, and many of the Catalan citizens are raised bilingually.

6.3.3 The AVI-unit decreases the number of errors

To determine whether the AVI-unit significantly decreases the number of errors in translations, we proposed testing the following hypothesis:

H_2 The use of the Post Edition AVI-unit significantly decreases the number of Total Error Points in a translation.

As discussed in section 6.3.2 we noted how the difference between the two language experiments are not significant, meaning that both the Spanish and Catalan datasets are subject to being treated as one dataset. In addition, it is important to understand that the comparison between the Post Edition and Post Edition improvement phase is a case of paired sampling (Snedecor & Cochran, 1989), where each data point in the first sample is uniquely linked to the data point in the second sample. It resembles a pre-test and post-test study which measures the impact of a specific alteration (in our case the use of the verification and providing of feedback mechanism). Because of this coupling the Post Editor that was linked to the Post Editor improver identified as being an outlier was removed. This means the dataset used for this hypothesis consists of 23 samples. As discussed in chapter 5.3.1 the quality of our translations are measured in terms of TEPs. This measurement is based on the number of errors created and their associated severity level (appendix D, E). The calculation for the TEPs was discussed in equation 5.

For both the Post Edition (table 17), as well as the Post Edition Improvement (table 18) phase we identified what errors were created and how severe they were. In addition the TEPs were calculated and we provide the percentage to see what proportion of errors fall in which error category. For table 18, we also include the delta (Δ), or difference in TEPs per error category and a percentage to see the decrease in TEPs for that specific error category. The error categories 'country standards' and 'miscellaneous' had no errors and are thus not depicted.

Table 17: Total Error Points found after the Post Edition phase

Error Category	Sev 1	Sev 2	Sev 3	Sev 4	TEP	Percentage
Accuracy	1	11	29	37	92,4	19,85%
Terminology	0	9	22	133	150,8	32,39%
Language Quality	0	0	24	160	156,8	33,68%
Style	0	0	17	53	62,8	13,49%
Formatting	0	0	1	2	2,8	0,60%
Totals	1	20	93	385	465,60	1,00

Table 18: Total Error Points found after the Post Edition Improvement phase

Error Category	Sev 1	Sev 2	Sev 3	Sev 4	TEP	Percentage	Δ	Decrease
Accuracy	3	4	22	38	82,8	24,35%	9,6	10,39%
Terminology	0	3	15	83	90,4	26,59%	60	40,05%
Language Quality	0	0	9	139	122	35,88%	35	22,19%
Style	0	0	13	35	43,6	12,82%	19	30,57%
Formatting	0	0	1	0	1,2	0,35%	1,6	57,14%
Totals	3	7	60	295	340,00	1,00		

Based on these results the first observation we make are to do with the distribution of errors within the error categories. Where most TEPs can be found in the 'Language Quality' category, which consist of errors such as grammar, spelling and readability. The categories 'Terminology' and 'Style' are related to company styleguides, Industry Guidelines and other factors related to the specific market that is being translated for. Combined these industry related factors form the 2nd major cause for created TEPs. The third category are the 'Accuracy' TEPs, which are related to inconsistencies, mistranslations and omissions/additions (appendix D for the full lists). Formatting TEPs are negligible.

To give a first idea on the improvement and distribution of the TEPs of the Post Edition (line Y) and Post Edition Improvement (line X) phases figure 26 shows a combined graph and frequency chart. The graph shows how the shift in TEPs found is moving towards the optimum of zero. The increased density of the Post Edition Improvement line is a logical effect of this improvement in combination with the maximum TEPs for the Post Edition Improvement phase being lower as the Post Edition phase. For the detailed experiment results study appendix M.

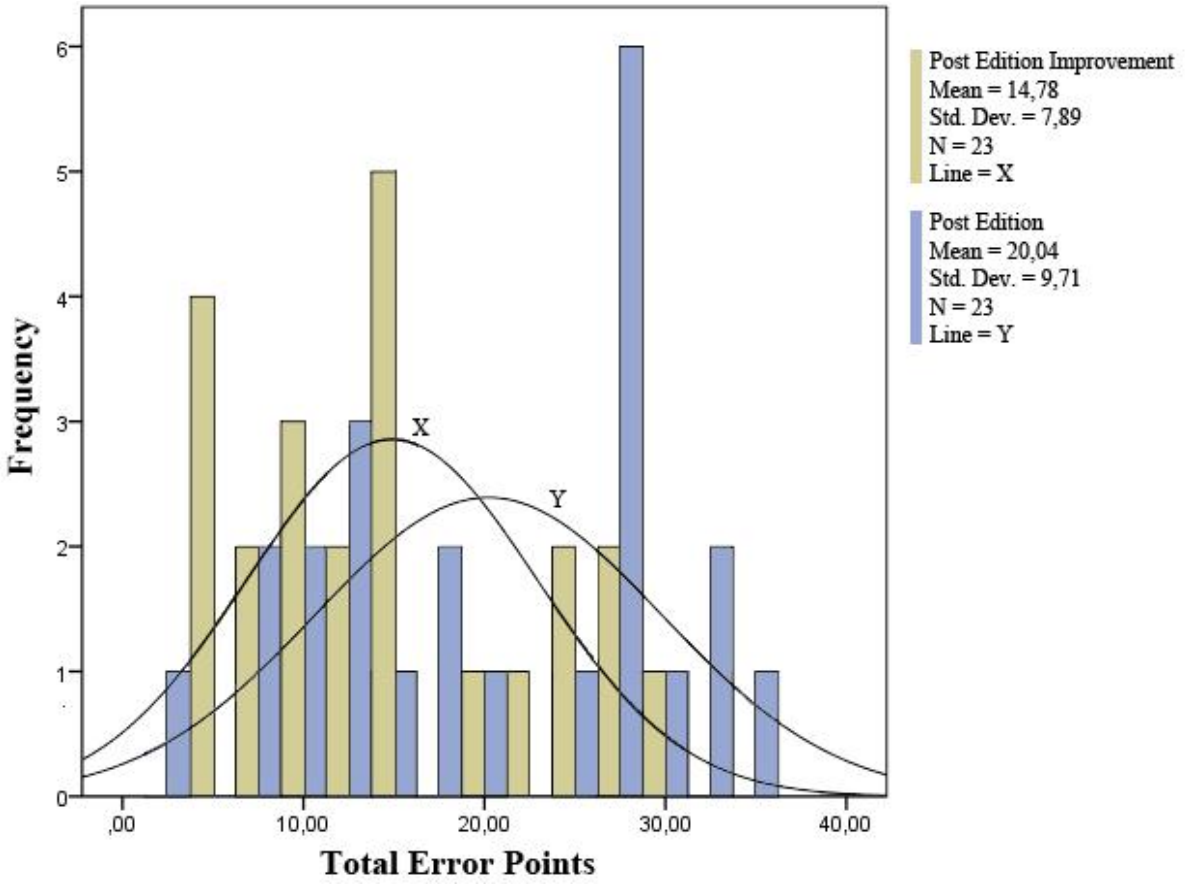


Figure 26: Post Edition and Post Edition Improvement frequency diagrams and normal distribution plot

To test whether the decrease in errors is significant we use a paired samples t-test (Snedecor & Cochran, 1989). In the Paired samples t-test we measure the difference the use of the AVI-unit and feedback mechanism had. The result for the t-test ($t(22) = 4{,}197, p < 0.000$) indicates a significant difference between the TEPs after the Post Edition phase and of the TEPs after the Post Edition Improvement phase. These results show that by adding three verifiers and having a second user improve that work based on error indications and feedback improves the quality of this crowdsourced task.

6.3.4 The quality output of the Post Edition AVI-unit

To determine whether the quality deliver by the AVI-unit is satisfactory, we define the following hypothesis:

H_3	The use of the Post Edition AVI-unit for a 350 word text delivers translations with Total Error Points of 5.3 or less.

As the number of TEPs provide only a numerical value of the results, we repeated the verbal quality scale and their judgment criteria (figure 27) to provide more context on the following results.

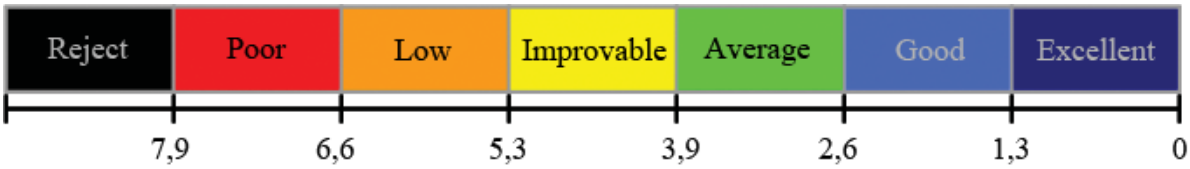


Figure 27: Total Error Points and the respective verbal quality scale (between Reject and Excellent) for a 350 word text (Schiaffino & Zearo, 2006).

The results for the users combined and the users separated per rank are shown in table 19. When looking into the mean average TEPs for the Post Edition Improvement phase we can see how neither of the three groups reach the threshold of 5,3 TEP. We can see how A ranked users nearly reach the verbal quality scale of "poor" with 8,1 TEPs and how the B and C rank users are all rejected in terms of translation quality. When we look into individual cases we see how only three A ranked users reach the TEP of 5,3 or lower (appendix M).

Table 19: Quality results for the different groups separate and combined

Group	Post Edition TEP	Post Edition Improvement TEP	Verbal Quality Scale	N
A users	13,3	8,1	Reject	8
B users	22.9	17,4	Reject	8
C users	26,1	22,6	Reject	8
Combined	20,0	14,78	Reject	24

Our results in general show that all but three (13%) translation would be rejected. Comparing these results to other crowdsourcing frameworks such as those from Ambati, Vogel and Carbonell (2010) and Zaidan and Callison-burch (2011) is difficulty. Especially, as our

experiment is being judged by a professional rating system, while using professionals to judge the translations. While other studies use automatic translation tools, or fail to describe the means of evaluating the translations. This means that we reject this hypothesis and note that in the current form the AVI-unit does not deliver translations with a TEP of 5,3 or less.

However, this conclusion does come with a but, as while analyzing the data we found a number of factors having a large influence on these results. In the next section we describe each of these factors and discuss their implications.

Uneditable sentences

The first remark we need to make is the following constraint: The system is designed in a way where users in the Post Edition Improvement phase can only edit sentences which have been indicated by *verifiers* to contain errors. This had an impact on the results in the following matter. Out of 499 errors the professional translators identified (severity not calculated) 25 were never found by crowdsourcing verifiers. Leading to the fact that these could never be improved by the Post Editor improvers. However, this factors has a rather small impact on the overall result, since these 25 errors represent approximately 5% of the errors.

Error overdetection

In addition, out of 597 sentences translated, the crowd workers commented (many users also commented that a sentence was "ok" and/or identified errors on 196 sentences where professional translators did not find any errors. The creation of a comment or identification of an error led to the display of this sentence to the Post Edition improver which in 45 cases led to the creation of errors, that were not there before. A stunning example of this can be seen in the case of user 8 in the Catalan experiment (appendix M), where user 20 introduced several new errors, reducing the quality of the text from being 'Very good' to 'Reject'. This seems to underline the subjectiveness of translations, and shows how the pressure of two or three people, unrightfully, stating something is wrong possibly led to this person injecting new errors.

High error occurrence

Third, we looked into the types of errors that were created and analysed whether this had any influence on the results. This study resulted in an interesting overview of identical errors (not calculating the severity) which had a very high occurence. In this overview (table 20) seven errors were indicated which had an occurences of 10 and more. They accounted for 210 out of

499 errors in the *post edition* phase and 168 out of 365 errors in the *post edition improvement* phase. The errors have been discussed with three professional translators to determine the severity and reasons behind these mistakes.

Table 20: Overview of errors with a high occurrence

English Text	Language	Error Type	Occurence of error	
			Post Edition	Post Edition Improvement
distro	Error for both	Industry-standard terminology	30	24
to	Error for both	Grammar / Syntax	105	83
please contact	Error for Catalan	Grammar / Syntax	11	9
Installing SSL certificate	Error for both	CA Guidelines	20	17
technical support	Error for both	CA Glossary	21	14
MGT	Error for Spanish	Grammar/ Syntax	13	13
should be logged	Error for Catalan	Grammar / Syntax	10	8
Total			**210**	**168**

Explanation on high occurring errors

Distro - This is a case of specific terminology. The translators identified there is a tendency among experts to use English words in Catalan or Spanish. CA translators prefer however that the Catalan and Spanish translation is used. They agreed that the use of the English word in a sentence like this is not wrong per-se and note that it would be hard for amateurs to find the correct translation without additional resources. The translation is understood as “preferred terminology” and maybe should have been included in the style guide or glossary

To - This is a case of a specific grammatical error. According to CA guidelines they always want to see the grammatical function before a product name. They prefer to see the article[16] in the translation. For both Catalan and Spanish this means adding "de" (preposition[17]) and "el" (article) to the translation, which is shortened as “del”.

Installing SSL certificate - The CA styleguide states that, for conformity, that 'installing' in Catalan is written as "Instal·lació" instead of, the also correct, "instal·lant". In Spanish it is written as "Instalaci" instead of "Instalando"

[16] An article are words like "a", "an" and "the". For example: What *an* interesting book!
[17] A preposition are words like "of", "on" and "into". For example: Is Tom *in* this photograph?

Please contact - The CA preferred translation of "contact" is "contacte" instead of the commonly used "contacteu". The use of the word "contacteu" is not wrong. This information was provided in the glossary along with the text. They state that due to the high frequency of these words being translated they require it to be consistent.

Technical support - The CA preferred translation of "technical support" is "l'assistència tècnica" in Catalan, and "soporte t" in Spanish. The versions the crowdworkers used "suport tècnic" and "apoyo t" are not necessarily wrong, but do not fit the guidelines of CA. This information was provided in the glossary along with the text, but not studied well enough by the participants.

MGT - MGT is the abbreviation of Metering Gateway which is still MGT in Spanish. However, many crowdworkers have not realized that even when in English the adjectives "the" is omitted, in Spanish "la" it is still required to be added.

Should be logged - The word "S'hauria" is a loan translation[18] and they prefer the use of the word "S'ha de registrar". They consider this style in a way that translators need to be educated on using this as they translate. A new or junior translator may make this mistake but they indicate they have to learn that this is the style CA uses when judging translations. However, it is not a word that does not exist in the Catalan lexicon.

These results show the difference between a normal or good translation and a professional translation adhering to specific *company standards*, *guidelines* and *translation practices* which otherwise would not apply. For example, in total we can identify 269 errors in the post edition phase and 174 errors in the post edition phase have nothing to do with *semantics* or *grammar*, and thus are not wrong in these regards (these errors include the high occurring error types "Industry-standard terminology", "CA Guidelines" and "CA Glossary" but do *not* include the Grammar/Syntax high occurring errors). These type of errors have much to do with the crowdworkers knowledge on some specific issues, instead of a serious lack of knowledge regarding the source or target language. In addition, the type of errors that are created are relatively simple to prevent. For example by improved instructions on these issues and guidance throughout the system.

[18] A loan translation is created when the target language does not have a translation for a specific word. In general it means that separate words are translated and combined to create a new lexeme in the target language.

Quality excluding CA Guidelines and CA Glossary errors

To identify what impact the errors related to company standards and guidelines had we performed an additional analysis to identify the quality of the texts, excluding the error categories related to these issues ("Terminology" and "Style"). The results of the translation quality is depicted in table 21 and shows how the average for the A ranked users has reached the wanted quality of "improvable" and in total 9 (40%) users now deliver a text of "improvable" quality.

Table 21: Quality results for the different groups separate and combined excluding the error categories Terminology and Style

	Post Edition TEP	Post Edition Improvement TEP	Quality Scale
A users	8,4	5,1	Improvable
B users	12,7	10,8	Reject
C users	13,9	13,8	Reject
Combined	11,6	9.9	Reject

Concluding this section we state that the crowdworkers in the current form are not able to deliver quality of an "improvable" quality. Based on our analysis we do see how a great majority of the errors are related to issues that are easily prevented. Additional analysis shows how addressing these issues properly means reaching lower TEPs is not unthinkable.

6.3.5 The performance of verifiers in the AVI-unit

To measure the performance of the verifiers we look into a series of binary classification metrics, and test the following hypothesis:

H_4	Higher ranked verification users perform better on the binary classification metrics than lower ranked users.

In the analysis for the verifiers we calculated a set of descriptive metrics about the verifiers (table 22). These metrics, precision, recall, accuracy and specificity are explained in table 23. In total 69 verifiers have been included in the analysis, 12 for the ranks A, B and C.

Table 22: Binary classification metrics for crowdsourcing verifiers

Group	Metric	Precision	Recall	Accuracy	Specificity	N
All	**Average**	0,58	0,61	0,61	0,56	69
	Std deviation	0,21	0,28	0,12	0,26	69
A	**Average**	0,62	0,62	0,59	0,57	69
	Std deviation	0,17	0,26	0,12	0,23	69
B	**Average**	0,54	0,59	0,63	0,59	69
	Std deviation	0,24	0,25	0,14	0,27	69
C	**Average**	0,57	0,61	0,59	0,52	69
	Std deviation	0,21	0,34	0,12	0,28	69

Table 23: Binary classification metrics and their descriptions

Metric	Description
Precision	the probability that if the verifier says that the sentence contains an error, it actually contains one.
Recall	the proportion of all sentences containing errors that were actually identified by the crowdworker.
Specificity	The proportion of correct sentences identified as such
Accuracy	the proportion of the total number of predictions that were identified correctly by the crowdworker.

Overall the average metrics lie very close to each other, implying that there is not much difference in how well the users find errors. However, when looking at the Std. deviations we can see how the spread in identifying sentences correctly (specificity) for A users is 0,23, while for C users this lies at 0,28. In addition when looking at Precision we see how A users not only perform better, but also have a lower spread. Another interesting conclusion that can be drawn lies in the combination of the Recall and Specificity statistics.

For example, the recall for C users seems to be average (0,61), but the data is deceiving. If you look closely to the standard deviation we see an extraordinary high number (0,34). Meaning that a wide range of users indicated to have found no errors, or found many errors. This hypothesis is strengthened by looking at the correlation to the specificity metric. By doing a Spearman's Rank-Order Correlation test (Pearson, 1900), we found a strong negative correlation ($\rho(24) = -.785$, $p < .000$), which is far above the critical value of $\rho = -.344$ (critical values can be found in appendix L) for a $\alpha = .05$ two-tailed level of significance. This shows that C users either tend to find lots of errors, even if there are none to find, or they find a very low amount of errors, missing many errors that are present in the texts.

Classifying errors and commenting on errors

As indicated in section 6.3.1 verifiers had the possibility to indicate an error and comment on this error for every sentences translated by the Post Editor. To analyze whether verifiers identified the right type of errors we performed an additional analysis. Take the following example: A certain sentence, translated by user X contains a *punctuation* error. In the verification stage, this sentence would be evaluated by another crowdworker. Although he saw a problem in this sentence, it does not mean he ticked the box for the *punctuation* error. He might have, but he could also have ticked the box for another error? and perhaps he did not tick any box but only left a comment?

To determine whether this happens a lot or not we performed a new analysis where we apply error mapping (appendix F). In this new analysis we recalculated the Accuracy metric, but now take into account the specific error that was ticked by the user. This new analysis results in a lowering of the Accuracy to 0,20 (Std. deviation = 7) instead of the average of 0,61 as derived from table 23.

When explaining the cause for the lower accuracy when applying error mapping we make use of an example which we found numerous times in the crowd platform. The verifiers in this example examined the translation of an English sentence into Spanish (table 24) created by a Post Editor.

Table 24: Original Sentence and Translation

	Sentence
English Sentence	Metering Gateway Installation and Configuration
Translated Sentence	Instalación y configuración de la puerta de enlace de medición (MGT)

What we notice when looking at the results from the verifiers is the following (table 25 on the next page). Each one ticked one of 9 error types and left a comment explaining what they saw. When comparing the three comments its clear they all found the same error: The Post Editor forgot to use a capital letter 'P' for the application name. The comments vary in detail but deliver the same message. User three even left a reference to the glossary which stated that application names should start with a capital letter. However, their choice of error differ significantly. Based on the review of the professional translator we know the Post Editors text did not adhere to the glossary and created a terminology error. Only user three indicated this, while the others ended up identifying a *software options*- and *typographical* Error. It is likely they did not study the glossary as user three did, or they could not make a distinction between the error types and selected what they thought fit the error best.

Table 25: Errors identified by verifiers

Verifier	Selected Error	Comment
User 1	Software Options	Instalación y configuración de la Puerta de enlace de medición (MGT)
User 2	Typographical	"Puerta" should be written with capital letter.
User 3	Inconsistent Terminology	Since Metering Gateway is an application it should be Puerta de enlace de medición (MGT), the first word should begin with a capital letter, as it stands in the glossary.

These observations indicate two things. First the importance of leaving a comment. For the Post Edition improver it was clear that the capital letter was the mistake and he fixed it in the improvement phase. However, out of 640 errors found by verifiers, 230 were not

accompanied by a comment and only identified an error. For the Post Editor improvers this leaves them with an unclear idea what a verifier might have meant, possibly leading to unnecessary errors or errors that were not fixed. Second, indicating the right error type seems to be hard and difficult to do.

Based on this section we conclude that A and B ranked users perform better in the verification phase than C-class users. Slightly smaller is the difference between A and B users, with an exception regarding the precision of the work, where A users deliver higher quality. When looking into the error classification system it clearly needs work. In the current situation it is too complicated for users to identify the right type of error, probably because having knowledge on the exact definitions is required to make a good distinction. When looking into the subjectivity of language and errors it is worth investigating the use of a comment-only verification system, allowing the Post Edition improver to base his improvement on the received feedback only.

6.3.6 The impact verifiers have on fixing errors in the AVI-unit

To measure the impact verifiers have on fixing errors we proposed testing the following hypothesis:

H_5	Higher ranked verifier groups perform better on the fixing- and destruction rate metrics than lower ranked groups.

In the analysis on the performance of the verification groups, we structured the data according to the relationship the groups have on the fixing- and destruction rate (table 26). Based on these results we can find that the use of an AAA group of verifiers led to a larger decrease in errors then when using a group of CCC verifiers. In addition we see how the group of CCC verifiers led to a higher introduction of new errors in the final translation. As discussed earlier, this could be because a Post Editor improver might feel pressured to change a piece of text based on wrong feedback, purely because multiple people advice him to do so. We do see that not all 'bad' advice gets copied by the Post Editor improvers. As we can see in the specificity metric (table 23) there are still many errors tagged as incorrect (0,56 were correct, thus 0,44 was not), while the destruction rate is almost three times smaller. This means that most of the times, this false errors are either ignored by Post Editor improvers or they refer to aspects of the sentence that do not affect its correctness (for instance fluency improvements).

Table 26: Efficiency scores for the different verification groups

Groups	Metric	Fixing Rate	Destruction Rate	Performance Rate
AAA	**Average**	0,40	0,14	0,26
	Std deviation	0,20	0,10	
BBB	**Average**	0,39	0,18	0,21
	Std deviation	0,24	0,07	
CCC	**Average**	0,26	0,21	0,05
	Std deviation	0,14	0,04	
ABC	**Average**	0,37	0,09	0,28
	Std deviation	0,06	0,09	
	Average Mean	0,36	0,15	0,20

To calculate the performance rate of the different groups we subtract the errors created from

the number of errors fixed, creating a performance rate. This rate gives a clue on the reduction of errors, when comparing the Post Edition improvement phase to the Post Edition phase. An analysis of variance (ANOVA), using a post hoc Tukey test (Winer, Brown & Michels, 1991), yielded a significant result between the groups AAA to CCC ($F(2, 65) = 4.528$, $p < 0{,}025$ and ABC to CCC ($F(2, 65) = 4.528$, $p < 0{,}007$). These results (appendix L) show how the use of three C ranked users only slightly decreases the total number of errors in a translation and is therefore the worst pick and using a configuration of ABC or AAA users is clearly the best pick. The fact that a group of an A, B and C user provides this result gives us valuable information on how we can have B and C users learn and grow in the platform. We see that combining lower ranked users with higher ranked users does not mean the overall quality will go down, thus giving the crowdsourcer the opportunity to save cost, while providing a learning platform for lower ranked users. We do note that further costs could possibly be cut by reducing the number of verifiers.

6.4 Results Summary

For each of the hypotheses we provide a summary of the tested hypotheses and their results.

Hypothesis regarding the dataset and differences between languages (table 27)

Table 27: Summary of the results

Hypothesis	Description	Test Result
H_1	The total error points created, either in the Post Edition or the Post Edition Improvement phase, by Spanish and Catalan crowdworkers are not significantly different.	Statistical tests show that the Catalan and Spanish experiment do not differ significantly.

Hypotheses regarding the performance of the AVI-unit (table 28)

Table 28: Summary of the results

Hypothesis	Description	Test Result
H_2	The use of an AVI-unit significantly decreases the number of error points in a translation.	Statistical tests show that the use of an AVI-unit significantly decreases the number of Total Error Points in a translation.
H_3	The use of a single AVI-unit for a 350 word text delivers translations with total error points of 5,3 or less.	With the exception of three out of 23 cases, the AVI-unit does not yet create translations with quality rank of "improvable". Professional translators do indicate how many errors that prevent these scores from being reached are related to company guidelines, which normal user, specifically amateurs volunteering in this experiment would not know. Additional testing shows that when excluding these type of errors all the A ranked users now create text with a quality rank of 'improvable'. In total 9 out of 23 users meet the Total Error Points threshold of 5,3 or less.

Hypotheses measuring the performance of the verifiers in the AVI-unit (table 29)

Table 29: Summary of the results

Hypothesis	Description	Test Result
H_4	Higher ranked verification users perform better on the binary classification metrics than lower ranked users.	Based on analysis A and B ranked verifiers perform better in correctly finding errors. In addition, we identify that the classification mechanism for identifying errors is too complicated.
H_5	Higher ranked groups perform better on the efficiency metrics than lower ranked groups.	Statistical tests prove that the groups AAA and ABC are significantly better in helping decrease errors than the group CCC.

7 Discussion

To scope the experiment we decided to only verify the workings of a single iteration of the AVI-unit, being the post-edition phase. Due to this decision, and the fact that very little research into this area has been done, it is difficult to make projections on the quality after adding another reviewing phase. In addition, in the experiment the motivational aspects for participants were not the same as in the proposed crowdsourcing system. To draw participants to the experiment we provided an iPad 2 (64gb) to participants who completed the experiment. We found it unfair however to add a weight to their chance of winning the iPad, based on the quality of the work they delivered. There was in this case an incentive to participate but no extrinsic motivation to create high quality work. However, we were in contact with a great many of the participants and many were very motivated to participate in the experiment, indicating a high level of intrinsic motivation to participate.

In the experiment, we note how A ranked users deliver significantly higher quality work than C ranked users, begging the question whether a true open crowdsourcing platform where any worker with any level of experience can participate is possible. One could say, that limiting participation to only those that qualify for a certain test defies the true nature of crowdsourcing. Though if this is the case, is the question. If we look at either Howe's or Brabham's definition of crowdsourcing, they never truly advocate that crowdsourcing should be open to anyone and if we look at other initiatives, for example at CrowdFlower or Mechanical Turk, we see how these systems are also working with qualifications, and worker satisfactory ratings to select allow them to work on new tasks. Although early adopters of crowdsourcing perhaps had this utopia, where anyone could participate, in mind. We must realize that the impact the ability and motivation a human being has to do a job greatly influences the outcome. We see this segregation on capabilities in the traditional job market, and although these boundaries might fade a little using crowdsourcing, some will always remain.

8 Conclusion

In this research we created and tested a quality assurance method for complex crowdsourcing tasks. The AVI-unit we propose is an improvement of quality assurance systems proposed by Bernstein et. al. (2010) and Ambati, Vogel and Carbonell (2010). Unlike the previous mentioned, the AVI-unit does not discard verifier information and congregates this to provide the crowdworker with feedback to improve his work. This created a relationship pattern between workers which allows them to collaboratively improve the quality of translations. In addition workers are paid a fair wage for their tasks, stimulating both *intrinsically* and *extrinsically* motivated participants.

We tested the use of the AVI-unit by starting an experiment where users of different ranks (A, B and C) work to create professional quality translations. In total over 226 people completed the test to measure translation capabilities, out of these 120 were selected to participate in the experiment. In total 24 texts, 648 sentences and 8400 words have been translated and improved. Based on these results we conclude that the AVI-unit significantly improves the quality of the translation delivered before and after the verification and improvement phase. Results show that the use of a single iteration of the AVI-unit does not yet provide professional quality translations.

However, we do note that 13% of the translations are classified as being the required 'improvable' translations when using the industry quality standard (Schiaffino & Zearo, 2006). Second, professional translators indicated how a number of recurring errors were not necessarily erroneous, but did not adhere to their companies guidelines. Whether this implies that the translation quality could actually be judged higher is uncertain. The company requires translations to adhere to certain guidelines for a reason. On the other hand we can also conclude that these type of errors can easily be prevented by improving the instructions to users. If we decide to assume these company specific errors can be fixed, we see that nearly 40% of the texts reach the quality threshold. We also note that the highest ranked users (class A) on average all reach this quality threshold.

Statistical analysis on the different users in the verification phase show how users of the A and B rank show only mildly different results. Where A ranked users are slightly more precise than their peers. The C ranked users have a tendency to either indicate that almost all sentences contain errors, or that almost none do. When performing more detailed analysis we

conclude that crowdworkers are unable to identify the correct error types and require either additional training, or the system requires a new and improved approach to the verification phase. Possibly by only relying on comments and feedback, and do not focus on error types at all. When looking at the verification groups we see the same tendency, where groups consisting of C ranked users score worse than others.

The significant difference in quality of delivered work between users of rank A and C deliver begs the question whether every user is suitable for these specific processes. This is not surprising since any organization dealing with complex task will hire only trained and suitable personnel, an approach that should not be any different in a crowdsourced system. The definition of Howe (Howe, 2006) does not imply that anyone can participate in the task, but merely states that the task should be published in an open call. To date, systems like Mechanical Turk are experimenting with means to the provider of tasks to ask for certain qualifications before tasks can be performed.

The research question: "*What translation quality can different configurations of crowdworkers deliver by applying a crowdsourcing quality assurance mechanism for software localization?"* can be answered by stating that *a single iteration* of the AVI-unit does not yet create professional quality translations. However, based on the results we can conclude that by using applying more specific training and taking more time to educate our crowdworkers we can make improvements in the proposed approach. On another note, this research proposes a new quality improvement mechanism that significantly increases the quality of crowdworkers work. This mechanism, the AVI-unit, is one of the first in its kind where crowdworkers improve work without interference of the crowdsourcer. In addition, our research shows that being more selective in picking only medium to high ranked users will greatly increase the quality of the work delivered, especially considering the more consecutive phases are added, the number of errors will decrease.

Concluding when looking at the future of crowdsourcing, we believe it is a matter of time before improved quality assurance methods are created and crowdsourcing will become the new outsourcing and although it is doubtful that all employees will work via digital platforms, for a number of jobs the world might look a lot different a decade from now.

9 Future research

To reduce the number of errors in the system automated tooling should be considered for errors that are most commonly made by crowdworkers. In addition the use of learning modules could be introduced. Where based on errors the users make they are suggested specific trainings that will help them improve their quality ranks. This in addition could help users motivation to participate in the platform. This research could be extended by looking into what sentences and language constructions cause more problems than others.

The verification phase sees room for improvement. For example a system where users are required to identify specific translation rules in a the style guide or glossary, creating a relationship between error and translation rule. Other approaches could include better instructions regarding the error types or more lenient error categories. To further improve the judgment system approaches should be considered where users highlight sections in the text that contained the error.

More research should be performed into the motivational factors of such a crowdsourcing system. This includes looking into experiment in which specific intrinsic and extrinsic motivational factors should be proposed to the different crowd workers and the effects on quality should be measured.

Lastly the ethics of crowdsourcing require more attention. These systems can severely impact the lives of people who's tasks might be performed by crowdsworkers in the future. Although these people will be able to work in the crowdtranslate platform they *will* lose job security and many social benefits. In addition there might be a lack of incentive for crowdsourcers to invest in their crowdworkers, through training and education. This responsibility will solely fall to the crowdworker. Finding the implications for these factors, and looking for an ethical and fair solution to these problems is of high importance.

10 Food for thought

It is time for industry to start thinking about real-time and real-world crowsourcing solutions. Letting humans collaborate through computerized crowdsourcing applications will be a way to reduce costs and become a competitive advantage for all those companies adopting this new technology. In our dynamic world, the future employee would work on a freelance basis for multiple companies. Due to the increase in scalability of the global workforce the work efficiency can greatly be increased. Employers will no longer be tied to certain employees, even if there is no work for them. And at the same time it will become easier for employees to find a suiting job elsewhere. These jobs could be anywhere, as they are no longer limited by their geographical location.

11 System Recommendations

Based on our findings in this experiment, we summarize our recommendations for the implementation of the AVI-Unit in a real live platform in the following list:

- Establish mechanisms in the interface to help workers comply with the guidelines provided: online glossary, automatic recognition of terminology in the original text.
- Provide online proof-reading tools: spellchecker and grammar revision.
- Improve feedback from verifiers to Post Editors by requiring a comment.
- Reduce error misclassification by clarifying which error type take precedence when several ones may apply, simplify the classification or simply remove it completely and focus only on comments.
- Adapt the number of verifiers according to the quality (ranking) of the Post Editor to reduce the probability of error injection.

12 Acknowledgements

I would like to thank the following people:

Utrecht University (Supervision)

- Remko Helms (Assistant Professor)
- Marco Spruit (Assistant Professor)

CA Technologies

- Victor Muntes (Director Research Europe – CA Technologies)
- Patricia Paladini Adell (Vice President European Localization – CA Technologies)
- Marc Sole (Research Member – CA Technologies)
- Arafat Ahsan (Machine Translation Expert – CA Technologies)
- Ilaria Cusumano & Clemens Bieg (Translator – CA Technologies)
- Glenn Crossman, John Kane & Michael Stricklen (Senior Software Architects – CA Technologies)
- The European Localization Team - CA Technologies

UPC

- Joseph Larriba (Professor UPC)
- Jawad Manzoor (MSc student UPC)
- Andrea Gritti (MSc student UPC) CROWD
- The over 225 people who volunteered to do the translation test and the 120 people who did additional translation and verification tasks.

13 References

Adda, G., & Cohen, K. B. (2011). Last Words Amazon Mechanical Turk : Gold Mine or Coal Mine ? *Computational Linguistics*, 37(2), 413–420.

Ambati, V., Vogel, S., & Carbonell, J. (2010.). Active Learning and Crowd-Sourcing for Machine Translation. Language, *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 6(6), 62-65.

Anderson, D. P. (2004). BOINC: A system for public-resource computing and storage. *Proceedings of the 5th international workshop on Grid Computing*, 4–10.

Automatic Language Processing Advisory Committee. (1966). *Language and Machines: Computers in Translation and Linguistics*

Backus, J. W., Bauer, F. L., Green, J., Katz, C., McCarthy, J., Naur, P., Perlis, A. J., et al. (1960). Report on the algorithmic language ALGOL 60. *Numerische Mathematik*, 2(1), 106–136.

Backus, John W. (1959). The syntax and semantics of the proposed international algebraic language of the zurich acm-gamm conference. *Proceedings of the International Comference on Information Processing*.

Banerjee, S., & Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. *Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 6, 65–72.

Bar-Hillel, Y. (1963). Is information retrieval approaching a crisis. *American Documentation*, 14(2), 95–98.

Beebe-Center, J. G., & Miller, G. A. (1956). Some psychological methods for evaluating the quality of translations. *Mechanical Translation*, 3(3), 73–80.

Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., et al. (2010). Soylent : A Word Processor with a Crowd Inside. *UIST '10 Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 313–322.

Blumberg, B., Cooper, D. R., & Schindler, P. S. (2008). *Business Research Methods (2nd ed.).* Berkshire: McGraw-Hill Education.

Borst. (2010). *Understanding Crowdsourcing*. Erasmus University.

Brabham, Daren C. (2011). *Crowdsourcing – what it isn't*. Retrieved October 6, 2012, from http://dbrabham.wordpress.com/crowdsourcing/

Brabham, Darran C. (2008). Crowdsourcing as a Model for Problem Solving: An Introduction and Cases. Convergence: *The International Journal of Research into New Media Technologies*, 14(1), 75–90. doi:10.1177/1354856507084420

Buhrmester, M., Kwang, T., & Gosling, S. D. (2011). Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data?, *Perspectives on Psychological Science*, 6(1), 3–5. doi:10.1177/1745691610393980

CA Technologies. (2012). *Linguistic Issue Tracking System 2.0*. Retrieved from CA Technologies Intranet (Protected, available on request).

Campbell, D. T., & Stanley, J. (1963). *Experimental and Quasi-Experimental design for research (1st ed.).* Belmont: Wadsworth Publishing.

Chandler, D., & Kapelner, A. (2010). Breaking Monotony with Meaning : Motivation in Crowdsourcing Markets. *University of Chicago, Mimeo*.

Charness, G., Dejong, D., Dufwen-, M., Fehr, E., Kirchsteiger, G., Rabin, M., Rosenthal, R., et al. (2000). *Pay enough or don't pay at all.*

Chomsky, N. (1956). Three models for the description of language. *Information Theory, IRE Transactions on*, 2(3), 113–124.

CommonSenseAdvisory. (2011). *The Language Services Market: 2011* (p. 75).

CommonSenseAdvisory. (2012). *Translation Future Shock* (p. 23).

Datar, M., Gionis, P., Indyk, P., Motwani, R. (2002). Maintaining stream statistics over sliding windows, *Society for Industrial and Applied Mathematics*, 31(6), 635-644.

Deci, E. L., Koestner, R., & Ryan, R. M. (2001). Extrinsic Rewards and Intrinsic Motivation in Education: Reconsidered Once Again. *Review of Educational Research*, 71(1), 1–27.

Dempser, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Royal Statistical Society*, 39(1), 1–38.

Denkowski, M., Al-haj, H., & Lavie, A. (2010). Turker-Assisted Paraphrasing for English-Arabic Machine Translation. *Computational Linguistics*, (June), 66–70.

Denkowski, M., & Lavie, A. (2010). Exploring Normalization Techniques for Human Judgments of Machine Translation Adequacy Collected Using Amazon Mechanical Turk. *Computational Linguistics*, (June), 57–61.

Doddington, G. (2001). Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics, *HLT '02 Proceedings of the second international conference on Human Language Technology Research*, 138–145.

Eisenberger, R., & Armeli, S. (1997). Can salient reward increase creative performance without reducing intrinsic creative interest? *Journal of Personality and Social Psychology*, 72(3), 652–663.

Elhadad, N., & Sutaria, K. (2007). Mining a Lexicon of Technical Terms and Lay Equivalents, *BioNLP '07 Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, 49–56.

Estelles-Arolas, E., & Gonzalez-Ladron-de-Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information Science*, 38(2), 189–200.

European Commission (2005). *Europeans and Languages*. Retrieved October 22, 2012, from http://ec.europa.eu/public_opinion/archives/ebs/ebs_237.en.pdf

Friedman, T. L. (2005). *The world is flat: A brief history of the twenty-first century*. New York: Farrar, Straus and Giroux.

Fromkin, V., Rodman, R., & Hyams, N. (1998). *An introduction to language* (7th Editio., p. 566). London: Thomson Wadworth.

Fuller, J., Jawecki, G., Muhlbacher, H., Füller, J., & Mühlbacher, H. (2007). Innovation creation by online basketball communities. *Journal of Business Research*, 60(1), 60–71.

Gao, Q., & Vogel, S. (2010). Consensus versus Expertise : A Case Study of Word Alignment with Mechanical Turk. *Computational Linguistics*, (June), 30–34.

Grady, C., & Lease, M. (2010). Crowdsourcing Document Relevance Assessment with Mechanical Turk, *CSLDAMT '10 Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 172–179.

Grubbs, F. (1969). Procedures for Detecting Outlying Observations in Samples, *Technometrics*, 11(1), 1-21.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). *Design science in information systems research*. MIS Q., 28(1), 75–105. Retrieved from http://dl.acm.org/citation.cfm?id=2017212.2017217

Heyman, J., & Ariely, D. (2004). Effort for payment. A tale of two markets. *Psychological science*, 15(11), 787–93.

House, J. (2001). Translation Quality Assessment: Linguistic Description versus Social Evaluation. Meta: *Journal des traducteurs*, 46(2), 243.

Howe, J. (2006). *The Rise of Crowdsourcing*. North, 14(14), 1–5. Retrieved June 11, 2012 from http://www.clickadvisor.com/downloads/Howe_The_Rise_of_Crowdsourcing.pdf

Hutchins, W. J. (1995). *Machine translation: a brief history*, 431–445.

Ipeirotis, P. G. (2010). analyzing the amazon Mechanical turk Marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2), 16–21.

Jurafsky, D., & Martin, J. H. (2008). *Speech and Language Processing* (2 edition ., p. 1024). New York: Pearson Prentice Hall.

Kaufmann, N., & Veit, D. (2011). More than fun and money . Worker Motivation in Crowdsourcing – A Study on Mechanical Turk, (2009), 1–11.

Kincaid, J. P. (1975). *Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel* (p. 49). Springfield.

Kittur, A., Chi, E. H., & Suh, B. (2008). Crowdsourcing user studies with Mechanical Turk. *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems* - CHI '08, 453.

Koehn, P., & Monz, C. (2006). Manual and Automatic Evaluation of Machine Translation between European Languages. *Proceedings of the Workshop on Statistical Machine Translation*, 102–121.

Leimeister, J. M. (2010). Collective Intelligence. *Business & Information Systems Engineering*, (4), 245–248.

Lyons, J. (1981). *Language and Linguistics: an Introduction. Cambridge*, UK: Cambridge University Press.

Localization Industry Standards Association (2003). *LISA QA metric*. Retrieved March 16, 2013, from http://producthelp.sdl.com/SDL_TMS_2011/en/Creating_and_Maintaining_Organizations/Managing_QA_Models/LISA_QA_Model.htm

Macklovitch, E. (1995). TransCheck--or the Automatic Validation of Human Translations. *Proceedings of the MT Summit V*.

Madsen, M. W. (2009). *The Limits of Machine Translation*.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15(4), 251–266.

Mason, W., Street, W., & Watts, D. J. (2009). Financial Incentives and the "Performance of Crowds", *HCOMP '09 Proceedings of the ACM SIGKDD Workshop on Human Computation* , 77-85.

Melamed, D. I., Green, R., & Turian, J. P. (2003). Precision and Recall of Machine Translation. NAACL-Short '03 Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technolog , 61–63.

Merriam-Webster. (2008). *Merriam-Webster's Collegiate Dictionary* (11th ed., p. 1664). Springfield: Merriam-Webster.

Murray, D. G., Yoneki, E., Crowcroft, J., & Hand, S. (2010). The case for crowd computing. *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds - MobiHeld '10* (p. 39). New York, New York, USA: ACM Press.

Negri, M., Bentivogli, L., & Marchetti, A. (2011). Divide and Conquer : Crowdsourcing the Creation of Cross-Lingual Textual Entailment Corpora. *Computational Linguistics*, 670–679.

Nov, O. (2007). What motivates Wikipedians? *Communications of the ACM*, 50(11), 60–64.

Oleson, D., Sorokin, A., Laughlin, G., Hester, V., Le, J., & Biewald, L. (2011). Programmatic Gold : Targeted and Scalable Quality Assurance in Crowdsourcing, *AAAI Publications, Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 43–48.

OpenSourceAlliance. (2012). *OpenSource Definition*. Retrieved May 14, 2012, from http://opensource.org/docs/definition.php

Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2002). BLEU : a Method for Automatic Evaluation of Machine Translation. *Computational Linguistics*, 311–318.

Pearson, K. (1900). X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine Series* 5(50), 157--175.

Pitler, E., & Nenkova, A. (2008). Revisiting Readability : A Unified Framework for Predicting Text Quality. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 186–195.

Provost, F., & Kohavi, R. (1998). Guest editors' introduction: On applied research in machine learning. *Machine Learning*, 30(2/3), 127–132.

Radford, A., Atkinson, M., Britain, D., Clahsen, H., & Spencer, A. (2009). *Linguistics: An Introduction (2nd editio., p. 450).* Cambridge University Press.

Reenskaug, T. M. (1979). *Models - Views - Controllers* (Technical note, Xerox PARC). Retrieved June 20, 2012, from http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf

Rogstadius, J., Kostakos, V., Kittur, A., Smus, B., Laredo, J., & Vukovic, M. (2011). An Assessment of Intrinsic and Extrinsic Motivation on Task Performance in Crowdsourcing Markets. *Fifth International AAAI Conference on Weblogs and Social Media*, 321–328.

Ross, J., Irani, L., Silberman, M.S., Zaldivar, A., & Tomlison, B. (2010). Who are the crowdworkers?: Shifting demographics in mechanical turk. *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, 2863-2872.

Schiaffino, R., Zearo, F. (2006). Developing and using a translation quality index. *Multilingual*, 53-58.

Schwarm, S. E., & Ostendorf, M. (2005). Reading Level Assessment Using Support Vector Machines and Statistical Language Models. *ACL '05 Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, 523–530.

Shah, S. K. (2006). Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development. *Management Science*, 52(7), 1000–1014.

Shapiro, S. S., Wilk, M. B. (1965). "An analysis of variance test for normality (complete samples)". *Biometrika* 52 (3-4): 591–611

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge, UK: Cambridge University Press.

Silberman, M. S., Irani, L., & Ross, J. (2010). Ethics and tactics of professional crowdwork. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2), 39.

Snedecor, G. W., & Cochran, W. G. (1989). *Statistical Methods, Eighth Edition*, Iowa State University Press.

Snow, R., Connor, B. O., Jurafsky, D., Ng, A. Y., Labs, D., & St, C. (2008). Cheap and Fast — But is it Good ? Evaluating Non-Expert Annotations for Natural Language Tasks. *Computational Linguistics*, (October), 254–263.

Society of Automotive Engineers (2000). Translation Quality Evaluation. *International Journal for Language and Documentation* (IJLD), 3, 24-25.

Tapscott, D., & Williams, A. D. (2007). *Innovation in the Age of Mass Collaboration*. New York, 8–10.

Wais, P., Lingamneni, S., Cook, D., Fennell, J., Goldenberg, B., Lubarov, D., Marin, D., et al. (2010). Towards Building a High-Quality Workforce with Mechanical Turk. *Computational Social Science and the Wisdom of Crowds*, 1–5.

Weaver, W. (1949). Translation. Machine translation of languages: fourteen essays (pp. 15–23). Cambridge, Massachusetts: Technology Press of M.I.T.

Wilks, Y. (2009). *Machine Translation: Its Scope and Limits*. New York: Springer.

Winer, B.J., Brown, D.R. & Michels, K.M. (1991). *Statistical principles in experimental design* (3rd ed.). New York: McGraw Hill

Zaidan, O. F., & Callison-burch, C. (2011). Crowdsourcing Translation : Professional Quality from Non-Professionals. *Computational Linguistics*, 1220–1229.

von Ahn, L. (2005). Human Computation, *Design Automation Conference*, 46, 418-419

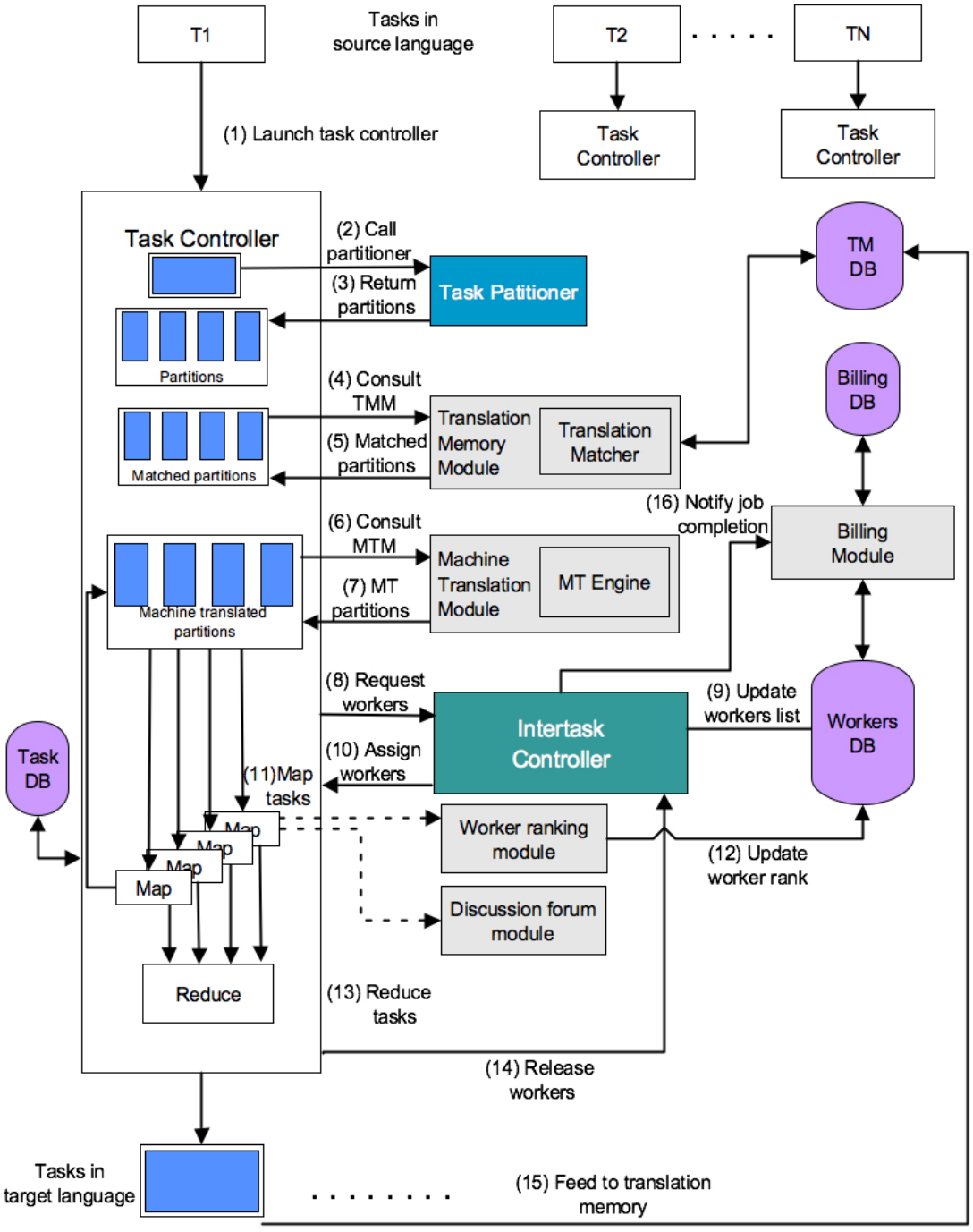
von Ahn, Luis, Maurer, B., McMillen, C., Abraham, D., & Blum, M. (2008). reCAPTCHA: human-based character recognition via Web security measures. *Science* (New York, N.Y.), 321(5895), 1465–8.

von Krogh, G., & von Hippel, E. (2006). The Promise of Research on Open Source Software. *Management Science*, 52(7), 975–983.

Appendices

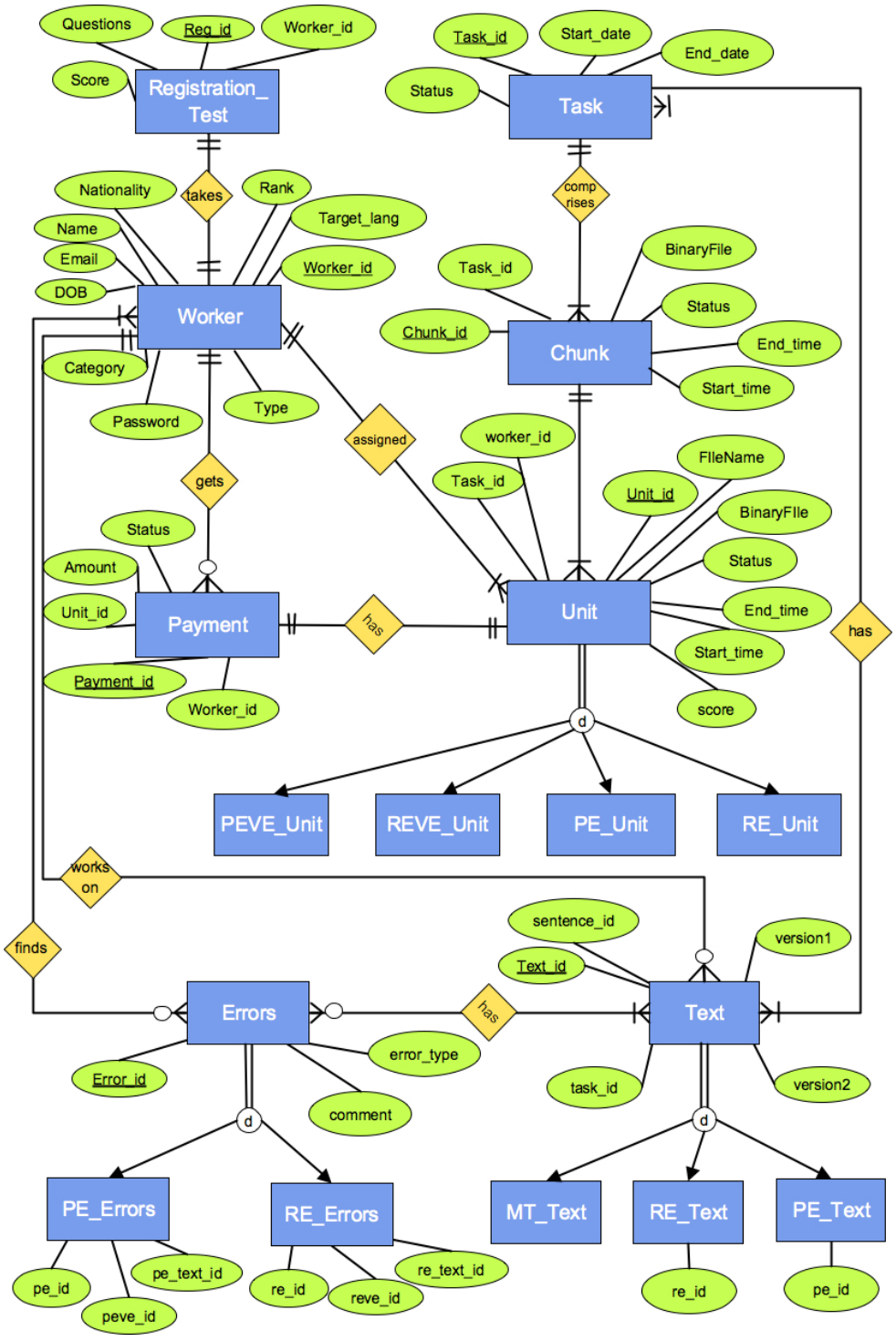
13.1 Appendix A: System Design: System Architecture

The crowdsourcing framework uses the following architecture consisting of 10 key modules. The Task Controller (TC), Task Partitioner, Translation memory module, Machine translation module, Map Controller (MC), Inter-task Controller (ITC), Working ranking module, Billing module, Registration module and Entry test module



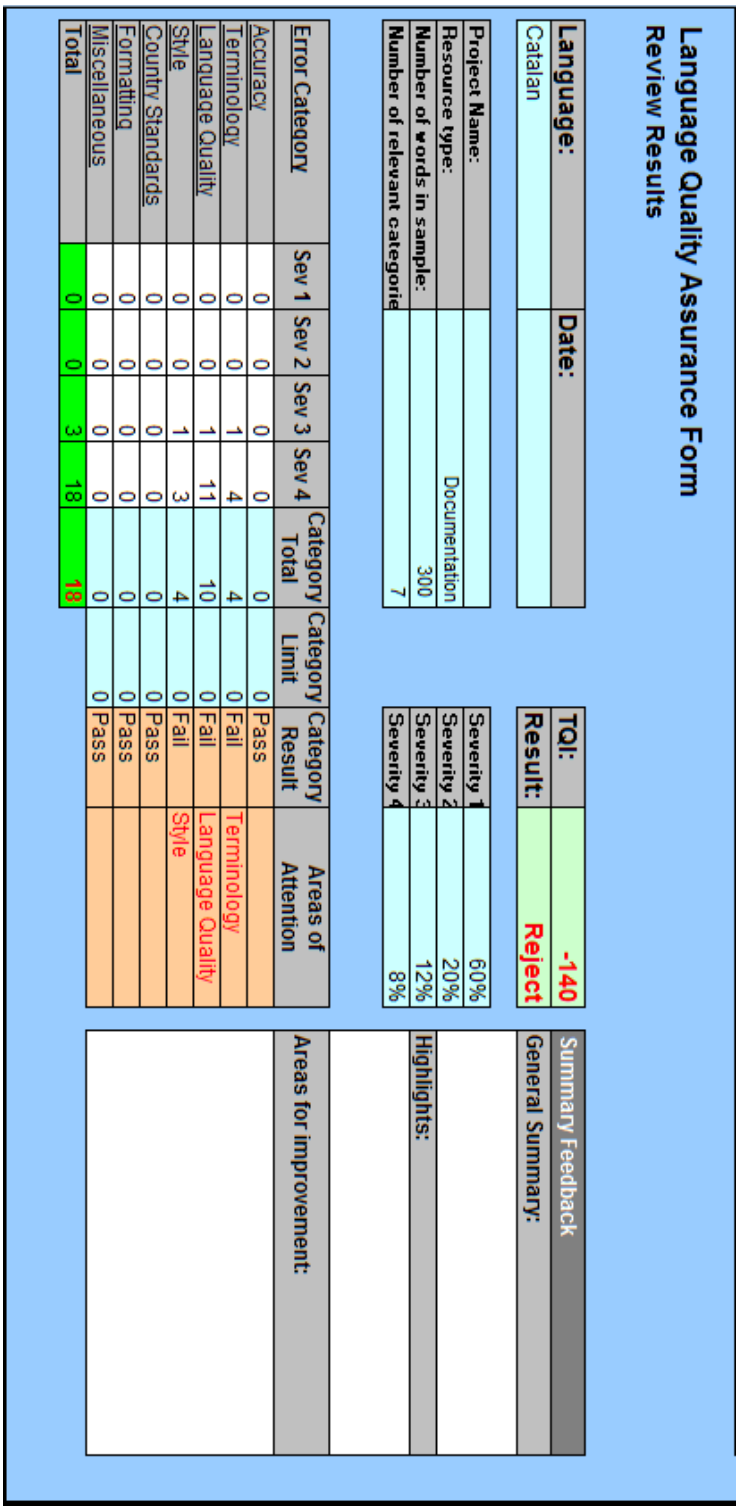
13.2 Appendix B:System Design: Entity Relation Diagram

The crowdtranslation platform ERD diagram revolves around the modules, Worker, Task and Unit. Each representing the different crowdworkers, including their ranking and payment information. Each task is linked to users through the different kind of unit instances representing Post edition, Reviewing, PE Verification and RE Verification tasks.



13.3 Appendix C:Translation Quality Review Sheet

The texts in the experiment are reviewed by professional translators using a method for reviewing the quality of texts. This method calculates the Total Error Points of a text based on having translators identify different types of errors and indicating the severity. Based on the size of the text and the number of errors the Translation Quality Index can be calculated.



Language Quality Assurance Form
Review Results

Language:	Date:
Catalan	

Project Name:	
Resource type:	Documentation
Number of words in sample:	300
Number of relevant categorie	7

TQI:	-140
Result:	Reject

Severity 1	60%
Severity 2	20%
Severity 3	12%
Severity 4	8%

Error Category	Sev 1	Sev 2	Sev 3	Sev 4	Category Total	Category Limit	Category Result	Areas of Attention
Accuracy	0	0	0	0	0	0	Pass	
Terminology	0	0	1	4	4	0	Fail	Terminology
Language Quality	0	0	1	11	10	0	Fail	Language Quality
Style	0	0	1	3	4	0	Fail	Style
Country Standards	0	0	0	0	0	0	Pass	
Formatting	0	0	0	0	0	0	Pass	
Miscellaneous	0	0	0	0	0	0	Pass	
Total	0	0	3	18	18			

Summary Feedback
General Summary:
Highlights:
Areas for improvement:

13.4 Appendix D: Translation Quality Error Categories and Types

The texts in the experiment are reviewed by professional translators using a method for reviewing the quality of texts. This method consists of classifying each error according to their Error Category and Error Type. An example would be the Error Type "Mistranslation" falling in the Error Category "Accuracy" (see below).

ERROR CATEGORIES

Examples

Accuracy	
Errors classified under the Accuracy category denote translation errors. They are normally detected by comparing the source and target texts.	
Mistranslation	The target language does not accurately reflect the meaning of the source text. This may include ambiguously or literally translated passages if the meaning of the orginal is lost or altered.
Omission/Addition	Source text information has been deleted from the target text, or information not found in the source text has been added to the target text.
Untranslated text	A portion of the source text has incorrectly been left untranslated (this does not include items left untranslated as per the project team's instructions).
Inconsistency	Terms or expressions are translated inconsistently throughout the text. This includes for instance headers or titles translated with a verb and then with a noun. (This section does not include terms found in the project or platform glossaries; such errors should be counted in the "Glossaries not followed" section)
Code defect	Any other defects introduced in software resources including (but not limited to) missing or extra delimiter (eg. double quote mark), swap of variables, etc.
Overtranslation	Over-translate variables in software or doc/help (this does not include items defined in instructions).
References	References to other sections or components of the product are incorrect, or references to third-party products are incorrect. This includes (but not limited to): Software references in the documentation and help, references to manual/chapter titles, addresses, phone numbers, links, cross-references, index references, graphics, and part numbers within or across components.
Completeness	Sections/content is missing in the target file.
Numeric	Numbers in target do not match English original
Trademarks	©,®, and ™ in target do not match the English original.

Terminology	
Errors classified under the Terminology category denote compliance errors. Usually, these are deviations from an approved translation glossary.	
Company glossary	The terminology used conflicts with Company/project and/or platform glossaries
Industry-standard terminology	The terminology does not follow generally accepted industry standards.
Inconsistent terminology	Inconsistent terminology within a product.

Language Quality	
Errors under the Language Quality category denote language errors. Usually, these are deviations from generally accepted language conventions.	
Grammar and/or syntax	The translation does not adhere to the target language-specific rules with regard to grammar or syntax.
Readability	The translation does not provide a concise and easy-to-understand narration manner based on target language style, though there is no grammer and spelling mistakes.
Spelling	The translation does not adhere to the target language-specific rules with regard to spelling. For Asian languages only, this includes not following proper spacing rules.
Capitalization or accentuation	The translation does not adhere to the target language-specific rule with regard to capitalization. If a capitalization error affects grammar, it should be considered a grammar error.
Hyphenation	The translation does not adhere to the target language-specific rule with regard to hyphenation. If a hyphenation error affects grammar, it should be considered a grammar error.
Typos	Misspellings and typographic errors.
Punctuation	The translation does not adhere to the target language-specific rule with regard to punctuation. If a punctuation error affects grammar, it should be considered a grammar error.

Style	
Errors under the Style Guide category denote compliance errors. Usually, these are deviations from an approved style guide or from the translation instructions.	
Company Guidelines	The translation does not adhere to company style guidelines and/or any other specifications provided for reference.
Project guidelines	The translation does not adhere to the instructions provided in the Translation Instructions, Questions & Answer file for the project.
Generic	Inappropriate level of formality, style conventions not followed, unidiomatic usage of target language. Note: Poor style may be difficult to describe in terms of individual errors. In that case it is possible to log one "generic" error, which should be supported by a number of examples.

Country Standards	
Errors under the Country Standards category denote localization errors. Usually, the translations do not conform to the standards in use in the target country.	
Target audience and/or market	Cultural references in the source text are not adapted to target audience or market. This may include any locale-specific reference to laws, regulations, job titles, proverbs, etc.
Date format	Date format is not adapted to country standards.
Unit of measures	Units of measure (e.g. length, weight, temperature, etc.) are not converted to country standards (unless any exceptional rules stated in the instruction).
Currency amounts	Currency is not converted to local currency.

Delimiters	Decimal delimiters are not adapted to country standards.
Addresses	Addresses are not adapted to country conventions.
Telephone numbers	Phone no are not adapted to country conventions.
Postal codes	Postal codes are not adapted to country conventions.
Shortcut keys	Shortcut keys in software not localized, or not following standards for the target language.

Formatting	
Errors under the Formatting category denote non-language errors. They are flagged only if they were the responsibility of the translator.	
Formatting	This category is mainly targeted at QAs performed on formatted files, i.e. the files are in the final format supplied to the end user.
File naming	Translation saved under incorrect file name.
Tags	Formatting tags changed.
Hidden text	Hidden text in RTF files translated.
Character styles	Character formatting incorrectly applied.
Font & font style	Text is incorrectly bolded, underlined, italicized, or in ALL CAPS. In software resources font or language code is not properly localized.
Table of Contents	Table of Contents is not updated to show the correct headings and page numbers.
Index	Index entries are not updated to match the English original.
Numbered lists	Numbered lists are incorrectly numbered.
Resizing	In software resource (mainly RC files) control size is not or improperly resized, hence the GUI truncation.
File format	Delivered file is not converted to required file format such as UTF-8, Unicode, escaping Unicode, etc.

Miscellaneous	
Any specific instructions from the client that do not fit any of the pre-defined categories above. Client specific errors must be specified here:	
	Any problem not listed in the above items and problems hard to categorize.

Repeat Errors	
Any repeated errors will not be punished with multiple occurrences. Instead, please log 1 issue into respective category.	
	All subsequent occurrences of the same error should be flagged as Repeat. Please count Repeat errors as 1 error. Rank according to the most serious occurrence of the mistake in question.

13.5 Appendix E:Translation Quality Severity Levels

The texts in the experiment are reviewed by professional translators using a method for reviewing the quality of texts. For each error the professional translator identifies it decides how severe this error is. The quality constraints per error are depicted below.

SEVERITY LEVELS

The quality of the translation as assessed by a Company Review Form depends on two factors: (a) the number of errors found in a given sample and (b) the severity of each error. When selecting a severity level, be as objective as possible. The examples below should help you select the appropriate severity level.

Severity Level	Examples
Severity 1	Errors leading to extreme consequence and errors that have been pointed out by the company as particularily severe for a particular project. Some examples of critical errors are: ● Errors in highly visible part of documentation or software, e.g. cover page, menu command. ● Error causing an application to crash or negatively modifying/misrepresenting functionality of the software. ● "Show stoppers", e.g. misrepresentations that may carry legal, safety, health, financial consequences. ● Error resulting in potentially offensive statements.
Severity 2	An error of a lesser severity than critical error. Very serious errors that jeopardize the meaning of a translated segment. Major errors are severe failures in accuracy, compliance, or language, as well as major errors that introduce extensive re-do or make-up job to slip project schedule. Some examples of major errors are: ● Accuracy errors that result in a significant change in meaning. "Significant" means that the user is very likely to be misled. ● Errors in visible part of documentation or software (header, TOC, chapter titles, help topic titles). ● Mistranslations resulting in misrepresentation of the source, e.g. omissions, misinterpretation of source, misleading statements. ● Query answers or previous QA feedback not applied. ● Grammar or syntax errors that are gross violations of generally accepted language conventions. ● Inconsistent terminology between GUI and Doc/Help leading to time-consuming modification.

Severity 3	Minor errors of a lesser severity than Severity 1 and 2, but should be fixed if at all possible Minor errors do not compromise the intelligibility of a translated segment. Some examples of minor errors are: ● Accuracy errors that result in a slight change in meaning. ● Small errors that would not confuse or mislead a user but could be noticed. ● Formatting errors (other than issues resulting in misrepresentation of source), e.g. wrong use of bold or italics. ● Wrong use of punctuation or capitalization not resulting in a loss of meaning. ● Generic error to indicate generally inadequate style (e.g. literal translation, "stilted" style, etc.) ● Grammar or syntax errors that are minor violations of generally accepted language conventions.
Severity 4	Minor errors with minimal impact to overall translation quality. These errors are going to be fixed to build up a PERFECT localization product if necessary. Some examples of minor errors are: ● Slightly visible format errors. ● Sporadic punctuation errors. ● Few occurrence of inconsistency in doc and help.

13.6 Appendix F: Platform Design: Error Mapping

CA professional translators identify errors according to the error system in Appendix D (chapter 13.4). In the crowd environment this complex judging system has been combined in a number of simplified errors. Each of these individual errors are linked to the complex error system according to the following mapping.

CA Errors		Crowd Errors
Accuracy		**Mapping**
Mistranslation	->	Mistranslation
Omission/Addition	->	Omission/ Addition
Untranslated text	->	Untranslatable text
Inconsistency	->	Software options
Code defect	->	Untranslatable text
Overtranslation	->	Omission/ Addition
References	->	Software options
Completeness	->	Omission/ Addition
Numeric	->	Mistranslation
Trademarks	->	Mistranslation
Terminology		
Company glossary	->	Inconsistent terminology
Industry-standard terminology	->	Inconsistent terminology
Inconsistent terminology	->	Inconsistent terminology
Language Quality		
Grammar and/or syntax	->	Grammar
Readability	->	Typographical error
Spelling	->	Typographical error
Capitalization or accentuation	->	Typographical error
Hyphenation	->	Typographical error
Typos	->	Typographical error
Punctuation	->	Punctuation
Style		
Company Guidelines	->	Style
Project guidelines	->	Style
Generic	->	Style

Country Standards		
Target audience and/or market	->	Style
Date format	->	Style
Unit of measures	->	Style
Currency amounts	->	Style
Delimiters	->	Style
Addresses	->	Style
Telephone numbers	->	Style
Postal codes	->	Style
Shortcut keys	->	Style

Formatting		
Formatting	->	Style
File naming	->	Style
Tags	->	Style
Hidden text	->	Style
Character styles	->	Style
Font & font style	->	Style
Table of Contents	->	Style
Index	->	Style
Numbered lists	->	Style
Resizing	->	Style
File format	->	Style

13.7 Appendix G: Platform Design: Post Editor and Reviewer ranking

Post Editors and reviewers work is being ranked after each task they complete. If the job contains little errors the quality of the work will be judged with a high score, if the job contains many errors, it will receive a low score. The scale for rating work is called the Translation Quality Index (TQI) first coined Schiaffino and Zearo (2006). The TQI (equation 8) is calculated based on two variables, being the Total Error Points (TEPs) found in the text (Total_Error_Points), and the number of errors which are allowed depending on the size of the text (Error_Allowance_Rate). The TQI is a standard used in the translation industry to determine the quality of vendor translations. The formula itself is designed with the rationale that in the translation industry a text seldom holds more than 5% errors in the text. This led to the creation of a scale where 5% errors equals a TQI of 0 and a perfect text equals a TQI of 100. However, the TQI can drop below 0 and reach a negative score when more errors are made. The multiply by 40 indicates that a sentence which contains as many errors as errors are allowed will reach a TQI of 60. A TQI which resembles an acceptable state of the text (**Error! Reference source not found.**). To calculate the TQI the Total_Error_Points and the rror_Allowance_Points need to be calculated first. How this works will be addressed in the coming section. Examples are included to provide insights on how the calculation works.

Equation 8: Calculating the Translation Quality Index

$$\text{TQI} = 100 - \text{round}\left(40\text{x}\ \frac{\text{Total\_Error\_Points}}{\text{Error\_Allowance\_Points}}\right)$$

The formulae to calculate the *error allowance* (Error_Allowance_Points) is depicted in Equation 9 and is based on the size of the text to be translated and the *error permitted rate* (Error_Permitted_Rate). The *Error Permitted Rate* is a rate set by the organization. The translation industry uses a percentage of 1.5% for manuals. This variable can differ based on the importance of the text to be translated. The *wordcount* is the total number of words in the sample to be tested.

Equation 9: Calculating the Error Allowance rate

$$\text{Error\_Allowance\_Points} = \text{Wordcount x Error\_Permitted\_Rate}$$

Example, part 1 of 3:

A text to be translated contains 800 words. The Error_Permitted_Rate rate is set at 1.5% for technical manuals. The Error Allowance Points are:

Error_Allowance_Points = 800 x 1.5% = 800 x 0.015 = 12

Calculating the TEPs in a text is more complicated and is not as straightforward as adding points for every error that is made. For a more comprehensive and precise judgment of the text the equation for the TEPs takes into account the severity of the error as well. The formula for calculating the TEPs in a text is depicted in equation 10. In total the system identifies 46 different error types from 7 different error categories (Appendix D). For each error that is found one of 4 different *severity percentages* has to be chosen (table 30).

Note: In the translation industry, a reviewer would indicate the severity of each error, a system which would be to time intensive for an online crowdsourcing system. As described in chapter 4.3.3.1 we propose a simplified method where the error type is linked to a severity score according to the weights in table 5. Where a weight of 1 is equal to 8% and a weight of 2.5 is equal to 20%. This means that the crowd verifier no longer has to choose between 46 different error types and 4 severities, but only has to choose between 1 of 7 *error categories*.

Equation 10: Calculating the Total Error Points for the TQI

$$\text{Total Error Points} = \sum_{Error\_Types}^{7} \left(10 \text{ x } \sum_{Severity\_Perc.}^{4} (Severity\_Percentage \text{ x } Error) \right)$$

Table 30: Error Categories, Types and Severity Percentages to calculate the TQI

Category	**Error Type**	**Severity** *(Category and Percentage)*							
		Cat.	Perc.	Cat.	Perc.	Cat.	Perc.	Cat.	Perc.
Accuracy	Mistranslation	1	*60%*	2	*20%*	3	*12%*	4	*8%*
	Omission/Addition	1	*60%*	2	*20%*	3	*12%*	4	*8%*
	...	1	*60%*	2	*20%*	3	*12%*	4	*8%*
Terminology	Company Glossary	1	*60%*	2	*20%*	3	*12%*	4	*8%*
	Industry Standards	1	*60%*	2	*20%*	3	*12%*	4	*8%*
	…	1	*60%*	2	*20%*	3	*12%*	4	*8%*

Language Quality	Grammar / Syntax	1	*60%*	2	*20%*	3	*12%*	4	*8%*
	Readability	1	*60%*	2	*20%*	3	*12%*	4	*8%*
	…	1	*60%*	2	*20%*	3	*12%*	4	*8%*
Style	Company Guidelines	1	*60%*	2	*20%*	3	*12%*	4	*8%*
	…	1	*60%*	2	*20%*	3	*12%*	4	*8%*
Country Standards	Target audience	1	*60%*	2	*20%*	3	*12%*	4	*8%*
	…	1	*60%*	2	*20%*	3	*12%*	4	*8%*
Formatting	Formatting	1	*60%*	2	*20%*	3	*12%*	4	*8%*
	…	1	*60%*	2	*20%*	3	*12%*	4	*8%*
Miscellaneous	Miscellaneous	1	*60%*	2	*20%*	3	*12%*	4	*8%*

Example, part 2 of 3:

In the 800-word text the following errors are found, two style errors of category 4 (thus 8% severity), a *grammar* error at category 2 (thus 20% severity) and a *software options* error of category 3 (thus 12% severity). The *total error points* in this text are as follows:

Total_Error_Points = round(10 x (8% x 2)) + round(10 x (20% x 1)) + round(10 x (12% x 1)) = 1.6 + 2 + 1.2 = 4.8

Finally, we provide a perspective for the TQI to give a sense of quality related to the scores. The TQI is standardized to an equivalent in terms of acceptability of the text. The terms are depicted in figure 28.

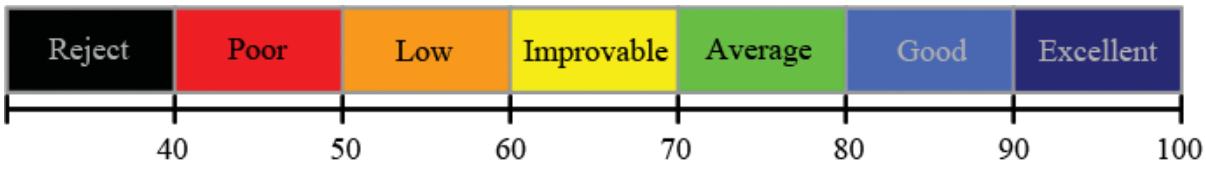


Figure 28: TQI scores and the quality of that text (Schiaffino & Zearo, 2006).

Example, part 3 of 3:

The total number of error points found in the text are 4. The *error allowance rate* was set to 12. We can now calculate the TQI.

$$TQI = 100 - round(40 \times (4{,}8 / 12)) = 87$$

With a TQI of 87 this text can be classified as being a *very good* translation and the user can add a Quality Index of 87 to his worker ranking.

13.8 Appendix H: Platform Design: Verifier ranking

Post edition verifiers and *reviewing verifiers* are ranked based on a separate ranking algorithm. Whenever a verifier identifies one of the nine error types (Table 3) in a sentence the sentence is flagged as *possibly erroneous* with a weight *w* equal to the rank of that verifier. To decide what errors are flagged as erroneous and which ones are discarded we identify the agreement of the users and take into account the *weight* they have. An error is defined as *definitely erroneous* when 50% or more of the sum of the *weight* indicates the error as *possibly erroneous*.

Example :

In a verification phase three users are added to verify a sentence. The users have respectively the user rankings of 90, 50 and 40 and thus have a combined weight of 180. Whenever the user with 90 indicates an error 50% or more of the sum of the combined weight is met and the error is defined as erroneous. If however the user with 50, or the user with 40 defines an error as erroneous respectively 28% and 22% of the sum of the combined weight is reached. As this is below the 50% threshold these errors would not be counted.

The formula to calculate verifier rank is based around the principle of comparing results with peers by taking into accounts the trustworthiness of the users. The *total error rate* (Total_Error_Rate) and the number of errors they have in accordance with the *total error rate*. The total verifier quality index (VQI) is capped at 100 points. To penalize eager beavers and lazy workers from not identifying errors or identifying errors that do not exist we introduce a penalty for deviating. Whenever a user would identify errors beyond a certain threshold from the *total error rate* this will result in penalty points. Each *excess deviation* (Excess_Devation) of more than three errors will lead to a penalty point multiplied by a *penalty rate* (Penalty_Rate) set at 4 points. The formula is depicted in equation 11.

Equation 11: Calculating the Verifier Quality Index

$$VQI = \text{round}\left(\left(\left(\frac{\text{Max\_Points}}{\text{Total\_Error\_Rate}}\right) x\ \text{Contribution}\right) + \left(\left(\frac{1}{\text{Total\_Error\_Rate}^3}\right) x\ 10^3\right) - (\text{Excess\_Deviation}\ x\ \text{Penalty\_Rate})\right)$$

Example:

In a text a verifier finds 9 possibly erroneous statements. Out of these 9 statements the system determines the total error rate to be 6. The verifier had conformity with 4 of these errors, and deviated on the 5 others. The verifying quality index for this task is calculated as follows.

$$VQI = ((100 / 6) * 4) + ((1 / 6^3) *10^3) - (2 \text{ x } 4) = 66.67 + 4.63 - 8 = 63$$

13.9 Appendix I: Platform Design: Calculating user rank over time

In appendix F and G we discussed how the quality of an individual task is ranked. In this section we discuss the how to calculate the overall rank of the user. The most convenient approach would be to average the quality indexes of all the users tasks and use that as the overall rank. This would however discriminate early work and early performance would haunt the worker. This is explained in the example below.

Example:

A fictional user "Peter" has performed 20 tasks over a period of time. His task scores are depicted in Figure 29 and show he has been improving significantly in the last 5 tasks. His average task scores are as follows:

- Peter scored an average of 54 over the past 20 tasks.
- Peter scored an average of 67 over the past 5 tasks.

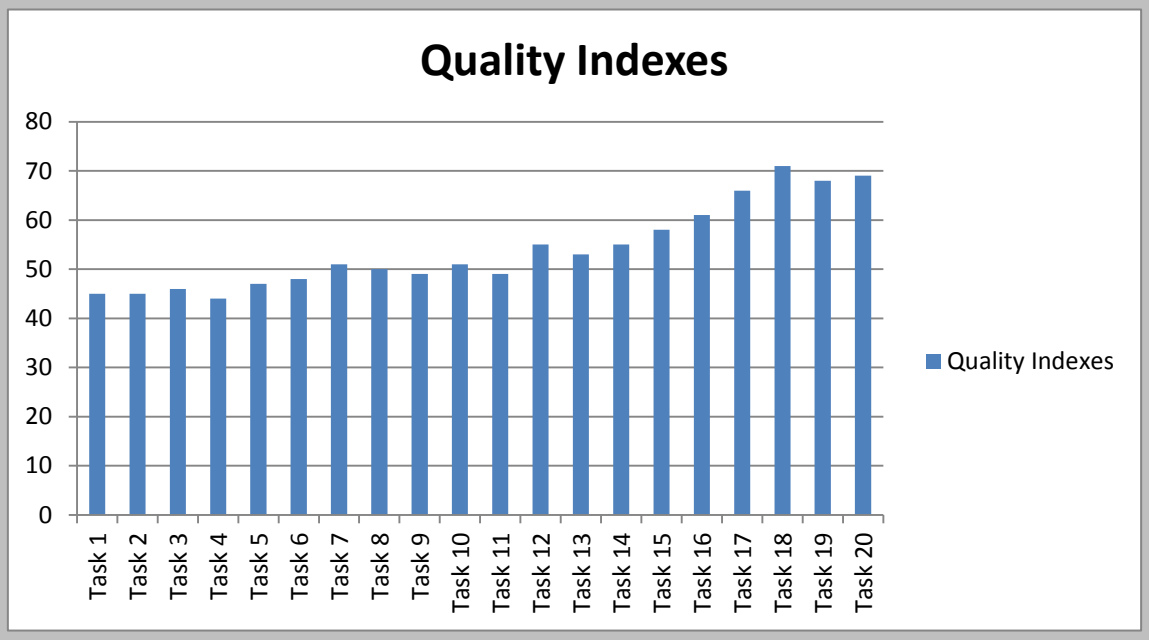


Figure 29: Quality indexes for a sample user

This example shows how using an approach to calculate Peter's average does not justify the work he has been delivering in the last few tasks (and thus would most likely deliver in the upcoming tasks).

To resolve this problem we propose the exponential histogram (Datar et. al, 2002). In an exponential histogram we use a differentiated weight of the work delivered based on the time the job was done. This approach is used to value work based over longer periods of time, without discriminating early work and taking into account improvements overtime.

The *crowd translation platform* uses this approach to determine ranking over time. In this exponential histogram we put the rank of the different jobs in *buckets*. Each bucket has a *timestamp* or *entry number* of the newest element and an array of the elements inside it. Whenever the *buckets* reach a certain date or entry number (a number identified by the *merge rate*) the last two buckets are compressed into 1 bucket, increasing the calculative weight of more recent elements.

- You choose a *time/entry window*. For example, 10 weeks before old entries are removed, or 20 entries before old entries are removed.
- A *merge rate*. For example, merge buckets after 3 new entries.

The exponential histogram will take a weighted average based on the following rules:

1) On new item *i*: Create new bucket of capacity 1 with timestamp = now, Average Quality Index = Quality Index
2) Merge buckets as soon as buckets of the same capacity > *merge_rate* and average the *quality index* of those buckets
3) Remove old buckets as soon as *timestamp_bucket – timestamp_now > time/entry window* or *number_of_tasks > time/entry window*.

Example:

Our user Peter has been performing translations for the past month. The known data to calculate the average according to the exponential histogram are stated below:

- Number of entries: 20
- Time/Entry window: 15 tasks
- Merge rate: 3 tasks

Example Continued:

Date	Quality Index	Date	Quality Index	Date	Quality Index	Date	Quality Index
16-Feb	45	12-Mar	48	12-Apr	49	4-May	61
17-Feb	45	14-Mar	51	14-Apr	55	11-May	66
22-Feb	46	14-Mar	50	15-Apr	53	13-May	71
26-Feb	44	29-Mar	49	2-May	55	16-May	68
6-Mar	47	9-Apr	51	3-May	58	19-May	69

Applying the rules of the exponential histogram to Peter's example would result in the following buckets and average quality index.

Date	Action	$[Bucket_{capacity}]_{Quality\ Index}$	Average QI
16-Feb	Add	$[B_1]_{45}$	45
17-Feb	Add	$[B_1]_{45}\,[B_1]_{45}$	45
22-Feb	Add	$[B_1]_{45}\,[B_1]_{45}\,[B_1]_{46}$	45
26-Feb	Merge, Add	$[B_2]_{45}\,[B_1]_{44}$	45
6-Mar	Add	$[B_2]_{45}\,[B_1]_{44}\,[B_1]_{47}$	45
12-Mar	Add	$[B_2]_{45}\,[B_1]_{44}\,[B_1]_{47}\,[B_1]_{48}$	46
14-Mar	Merge, Add	$[B_2]_{45}\,[B_2]_{45}\,[B_1]_{51}$	47
14-Mar	Add	$[B_2]_{45}\,[B_2]_{45}\,[B_1]_{51}\,[B_1]_{50}$	48
29-Mar	Add	$[B_2]_{45}\,[B_2]_{45}\,[B_1]_{51}\,[B_1]_{50}\,[B_1]_{48}$	48
9-Apr	Merge, Add	$[B_2]_{45}\,[B_2]_{45}\,[B_2]_{50}\,[B_1]_{51}$	48
12-Apr	Add	$[B_2]_{45}\,[B_2]_{45}\,[B_2]_{50}\,[B_1]_{51}\,[B_1]_{51}$	48
14-Apr	Add	$[B_2]_{45}\,[B_2]_{45}\,[B_2]_{50}\,[B_1]_{51}\,[B_1]_{49}\,[B_1]_{55}$	49
15-Apr	Merge,Merge,Add	$[B_3]_{47}\,[B_2]_{52}\,[B_1]_{53}$	51
2-May	Add	$[B_3]_{47}\,[B_2]_{52}\,[B_1]_{53}\,[B_1]_{55}$	52
3-May	Add	$[B_3]_{47}\,[B_2]_{52}\,[B_1]_{53}\,[B_1]_{55}\,[B_1]_{58}$	53
4-May	Delete, Merge, Add	$[B_2]_{52}\,[B_2]_{55}\,[B_1]_{61}$	56
11-May	Add	$[B_2]_{52}\,[B_2]_{55}\,[B_1]_{61}\,[B_1]_{66}$	59
13-May	Add	$[B_2]_{52}\,[B_2]_{55}\,[B_1]_{61}\,[B_1]_{66}\,[B_1]_{71}$	61
16-May	Merge, Add	$[B_2]_{52}\,[B_2]_{55}\,[B_2]_{66}\,[B_1]_{68}$	60
19-May	Add	$[B_2]_{52}\,[B_2]_{55}\,[B_2]_{66}\,[B_1]_{68}\,[B_1]_{69}$	62

We stated earlier what the quality indices would be if we used a normal averaging technique. Peter scored an average QI of 54 over the past 20 tasks.. Peter scored an average QI of 67 over the past 5 tasks.

With the use of the exponential histogram this would result in scoring a QI of **62**.

13.10 Appendix J: Experiment: Translation Capability Test

A *translation capability test* was designed to assess the rank (A, B or C) the user has in terms of translation capabilities. The test consisted of three parts. The first part consisted of a statement in English and multiple translations in the target language (Figure 30 31). In this seven question multiple choice segment the user had to select the correct translation from four options, which measures the ability of the user to identify higher quality sentences.

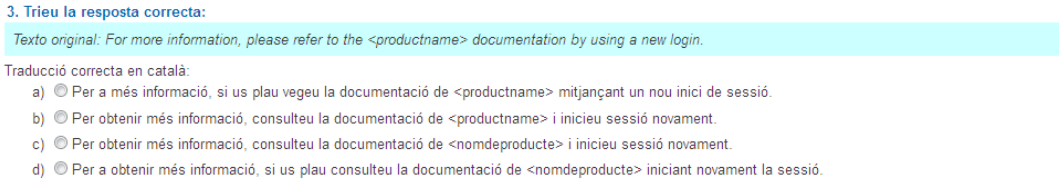


Figure 30: Registration test, question section 1

The second part of the test consisted of questions were the user had to identify errors in translations (figure 31). In this process, the user has to identify if and which errors can be found in the text. This emphasizes the difference between the type of errors that can be made and found in the text. Nine different error types were provided in the form of checkboxes. A statement could have no, one or multiple errors. This section consisted of ten questions.

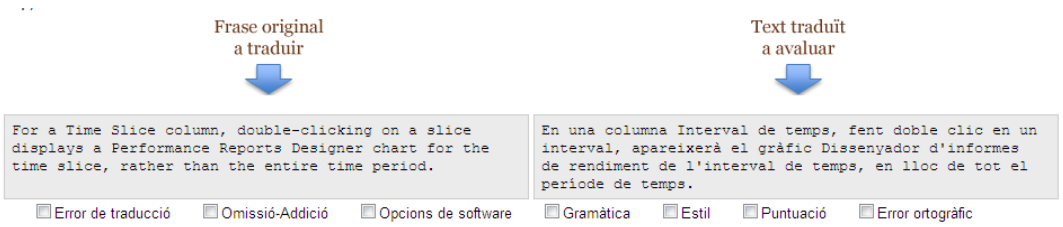


Figure 31: Registration test, example section 2

In part three, the users were given a set of nine statements in the target language only and they were to find grammar, style, punctuation, or typographical errors (figure 32).

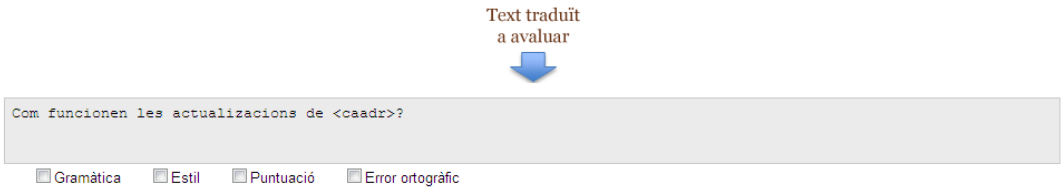


Figure 32: Registration test, example section 3

The test scores are calculated by adding a point for every correct question. Users get partial points for the verification questions if they get at least one type of error correct and no more than one wrong. Finally the points were summed and the scores are normalized on a 0 to 1 scale. The test was used to assign the user ranks A, B and C which is discussed in chapter 6.2.

13.11 Appendix K: Experiment: Technical Manual

Metering Gateway Installation and Configuration

The MGT application is a metering gateway that accepts metering data from grids within a customer's datacenter and forwards that data to <productname>'s metering system. In addition, MGT provides an interface through which the customer may manage the SSL certificate MGT uses to communicate with <productname>'s metering system.

Initial Application Setup

To install and configure the MGT application, perform the following:

1. Download the application and copy it to the <productname>'s distro.
2. From the <productname>'s distro, import the application to the grid.
3. Log in to your grid and resize the data volume. The size of the data volume should be sufficient to store at least 30 days of metering data for all grids that report to this instance of MGT (approximately 4 MB per grid for 30 days of metering data).
4. Configure the application. To retrieve the set of parameters that may be configured, see the MGT reference.
5. Start the application.
6. Setup the client- side SSL certificate

Note: The grid on which the MGT application is running must have access to the internet because MGT communicates with <productname>'s metering server.

Installing SSL Certificate

Follow these steps to install a signed SSL certificate on the Metering Gateway.

1. Start the Metering Gateway application. A message should be logged to the grid dashboard specifying the SSL certificate is missing.
2. Log in into the MGT application and create the SSL certificate.
3. Copy and paste the displayed certificate signature request into an e- mail and send it to technical support.
4. When technical support replies with the signed certificate, save the certificate to your local machine.
5. Verify the signed SSL certificate has been copied by logging into the MGT application.
6. Restart the MGT appliance.
7. Verify no error messages are logged to the grid dashboard. If there are error messages, please contact technical support.

13.12 Appendix L: Experiment: Statistical Tests

Independent T-Test for the Difference between the Catalan and Spanish test

Group Statistics

	Language	N	Mean	Std. Deviation	Std. Error Mean
Section 1	Catalan	102	,7983	,23301	,02307
	Spanish	114	,6842	,26504	,02482
Section 2	Catalan	102	,4036	,15062	,01491
	Spanish	114	,3383	,12163	,01139
Section 3	Catalan	102	,4341	,17389	,01722
	Spanish	114	,4372	,17549	,01644
Combined	Catalan	102	,5203	,14778	,01463
	Spanish	114	,4623	,13089	,01226

*Note: Continues on next page

Independent T-Test for the Difference between the Catalan and Spanish test

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Section 1	Equal variances assumed	2,726	,100	3,343	214	,001	,11411	,03413	,04683	,18139
	Equal variances not assumed			3,367	213,938	,001	,11411	,03389	,04731	,18091
Section 2	Equal variances assumed	5,439	,021	3,524	214	,001	,06537	,01855	,02881	,10192
	Equal variances not assumed			3,483	194,166	,001	,06537	,01877	,02835	,10238
Section 3	Equal variances assumed	,076	,782	-,129	214	,898	-,00307	,02382	-,05001	,04387
	Equal variances not assumed			-,129	211,769	,898	-,00307	,02380	-,04999	,04385
Combined	Equal variances assumed	,701	,403	3,059	214	,003	,05800	,01896	,02062	,09537
	Equal variances not assumed			3,038	203,110	,003	,05800	,01909	,02036	,09564

Shapiro-Wilk test for normality for the combined test scores for the Catalan and Spanish users

Tests of Normality

		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Language	Statistic	df	Sig.	Statistic	df	Sig.
Combined	Catalan	,068	102	,200*	,979	102	,109
	Spanish	,062	114	,200*	,979	114	,073

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Shapiro-Wilk test for normality for the Total Error Points for the Catalan and Spanish users

Tests of Normality

		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	Language	Statistic	df	Sig.	Statistic	df	Sig.
TEP after PE phase	Catalan	,193	12	,200*	,925	12	,328
	Spanish	,228	12	,085	,911	12	,219
TEP after PE improvement phase	Catalan	,179	12	,200*	,925	12	,331
	Spanish	,194	12	,200*	,913	12	,233

a. Lilliefors Significance Correction

*. This is a lower bound of the true significance.

Independent T-Test for difference of the means between Catalan and Spanish phases

Group Statistics

	Language	N	Mean	Std. Deviation	Std. Error Mean
PE after Post Edition	Catalan	12	14,2667	7,39992	2,13617
	Spanish	11	15,3455	8,89453	2,68180
PE after Post Edition Improvement	Catalan	12	18,4667	11,22961	3,24171
	Spanish	11	22,1818	7,80228	2,35248

*Note: Continues on next page

Independent T-Test for difference of the means between Catalan and Spanish phases

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
PE after Post Edition Improvement	Equal variances assumed	,856	,365	-,317	21	,754	-1,07879	3,40030	-8,15009	5,99251
	Equal variances not assumed			-,315	19,558	,756	-1,07879	3,42860	-8,24110	6,08353
	Equal variances assumed	3,072	,094	-,913	21	,372	-3,71515	4,06946	-12,17806	4,74776
	Equal variances not assumed			-,928	19,644	,365	-3,71515	4,00535	-12,07989	4,64959

Pearson's Correlation between Sensitivity and Recall classification metrics for

Descriptive Statistics

	Mean	Std. Deviation	N
Recall	,611133	,3398982	24
Specificity	,522243	,2839844	24

Correlations

		Recall	Specificity
Recall	Pearson Correlation	1	-,795**
	Sig. (2-tailed)		,000
	N	24	24
Specificity	Pearson Correlation	-,795**	1
	Sig. (2-tailed)	,000	
	N	24	24

**. Correlation is significant at the 0.01 level (2-tailed).

Tukey HSD Multiple-comparisons Anova to analyse the effieciency of the different groups.

ANOVA

Total-efficiency

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	,592	3	,197	4,528	,006
Within Groups	2,831	65	,044		
Total	3,423	68			

Multiple Comparisons

Dependent Variable:Total-efficiency

			Mean Difference			95% Confidence Interval	
	(I) Group_#	(J) Group_#	(I-J)	Std. Error	Sig.	Lower Bound	Upper Bound
Tukey HSD	1	2	,0470000	,0729643	,917	-,145389	,239389
		3	,2120000*	,0729643	,025	,019611	,404389
		4	-,0213333	,0729643	,991	-,213723	,171056
	2	1	-,0470000	,0729643	,917	-,239389	,145389
		3	,1650000	,0695688	,093	-,018436	,348436
		4	-,0683333	,0695688	,760	-,251769	,115103
	3	1	-,2120000*	,0729643	,025	-,404389	-,019611
		2	-,1650000	,0695688	,093	-,348436	,018436
		4	-,2333333*	,0695688	,007	-,416769	-,049897
	4	1	,0213333	,0729643	,991	-,171056	,213723
		2	,0683333	,0695688	,760	-,115103	,251769
		3	,2333333*	,0695688	,007	,049897	,416769

*. The mean difference is significant at the 0.05 level.

Pearson Correlation Critical Values

Table of critical values for Pearson correlation – N (not df) is in column 1

	One Tailed Probabilities			
	0.05	**0.025**	**0.005**	**0.0005**
	Two-Tailed Probabilities			
N	**0.1**	**0.05**	**0.01**	**0.001**
4	0.900	0.950	0.990	0.999
5	0.805	0.878	0.959	0.991
6	0.729	0.811	0.917	0.974
7	0.669	0.754	0.875	0.951
8	0.621	0.707	0.834	0.925
9	0.582	0.666	0.798	0.898
10	0.549	0.632	0.765	0.872
11	0.521	0.602	0.735	0.847
12	0.497	0.576	0.708	0.823
13	0.476	0.553	0.684	0.801
14	0.458	0.532	0.661	0.780
15	0.441	0.514	0.641	0.760
16	0.426	0.497	0.623	0.742
17	0.412	0.482	0.606	0.725
18	0.400	0.468	0.590	0.708
19	0.389	0.456	0.575	0.693
20	0.378	0.444	0.561	0.679
21	0.369	0.433	0.549	0.665
22	0.360	0.423	0.537	0.652
23	0.352	0.413	0.526	0.640
24	0.344	0.404	0.515	0.629
25	0.337	0.396	0.505	0.618
26	0.330	0.388	0.496	0.607
27	0.323	0.381	0.487	0.597
28	0.317	0.374	0.479	0.588

29	0.311	0.367	0.471	0.579
30	0.306	0.361	0.463	0.570
35	0.283	0.334	0.430	0.532
40	0.264	0.312	0.403	0.501
45	0.248	0.294	0.380	0.474
50	0.235	0.279	0.361	0.451
60	0.214	0.254	0.330	0.414
70	0.198	0.235	0.306	0.385
80	0.185	0.220	0.286	0.361
90	0.174	0.207	0.270	0.341
100	0.165	0.197	0.256	0.324
200	0.117	0.139	0.182	0.231
300	0.095	0.113	0.149	0.189
400	0.082	0.098	0.129	0.164
500	0.074	0.088	0.115	0.147
1000	0.052	0.062	0.081	0.104

13.13 Appendix M: Experiment: Individual Crowdworker Results

Catalan experiment

Post Edition				Verification		Post Edition Improvement		
Rank	User id	Total Error Points		Type		Rank	User id	Total Error Points
A	1	8,80	->	ABC	->	A	13	5,20
	2	12,00	->	AAA	->		14	8,00
	3	6,80	->	BBB	->		15	6,00
	4	10,80	->	CCC	->		16	14,40
B	5	12,80	->	ABC	->	B	17	9,20
	6	28,40	->	AAA	->		18	14,40
	7	33,60	->	CCC	->		19	15,20
	8	1,60	->	BBB	->		20	8,80
C	9	34,80	->	AAA	->	C	21	28,40
	10	18,00	->	BBB	->		22	15,60
	11	26,80	->	CCC	->		23	22,00
	12	27,20	->	ABC	->		24	24,00

Spanish Experiment

Post Edition				Verification		Post Edition Improvement		
Rank	User id	Total Error Points		Type		Rank	User id	Total Error Points
A	1	8,00	->	ABC	->	A	13	4,00
	2	18,40	->	AAA	->		14	5,20
	3	14,40	->	BBB	->		15	10,00
	4	27,20	->	CCC	->		16	12,00
B	5	28,40	->	ABC	->	B	17	12,80
	6*	32,40	->	AAA	->		18*	44,40
	7	26,00	->	CCC	->		19	19,60
	8	20,00	->	BBB	->		20	14,40
C	9	27,20	->	AAA	->	C	21	29,20
	10	13,20	->	BBB	->		22	8,40
	11	29,60	->	CCC	->		23	27,20
	12	31,60	->	ABC	->		24	26,00

***Note:** User id 18 of the Spanish experiment was an outlier and excluded from the results. The corresponding user 6 was therefore no longer useable.