På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extra-ordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnad som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© [Vamshi Prakash Kondapalli]
Institutionen för datavetenskap
Department of Computer and Information Science

Final thesis

Evaluation of the user interface of the BLAST annotation tool

By

Kondapalli Vamshi Prakash

LIU-IDA/LITH-EX-A−12/039--SE

2012-06-18
Final Thesis

Evaluation of the user interface of the BLAST annotation tool

By

Kondapalli Vamshi Prakash

LIU-IDA/LITH-EX-A – 12/039-- SE

2012-06-18

Supervisor: Sara Stymne
Dept. Of Computer and Information Science

Examiner: Lars Ahrenberg
Dept. Of Computer and Information Science
Acknowledgement

I would like to express my sincere gratitude to a number of people who have supported me in my work and contributed to this thesis.

This evaluation study was carried out under Human-Centered Systems division in the Department of Computer and Information Science at Linköping University. During my thesis work I took support of many people, which made to move a step further in evaluating this annotation tool called BLAST.

First of all, I would like to express my sincere thanks to my Supervisor Sara Stymne who has guided and supported me throughout the Thesis. Thank you for allowing me to work under your supervision.

I also want to express my respect and gratitude towards Lars Ahrenberg for allowing me to do a Master Thesis in the area of usability under Natural Language Processing Laboratory (NLPLAB).

I would also like to acknowledge Johan Åberg for providing useful information (Web links, Journals, Books) to carry out the Usability Evaluation.

I would like to express thankfulness to Mattias Arvola for giving valuable input on usability methods and issues.

Last but not the least; I would like to thank all my participants who showed their support and patience during evaluation study.
Abstract

In general, annotations are a type of notes that are made on text while reading by highlighting or underlining. Marking of text is considered as error annotations in a machine translation system. Error annotations give information about the translation error classification.

The main focus of this thesis was to evaluate the graphical user interface of an annotation tool called BLAST, which can be used to perform human error analysis for any language from any machine translation system. The primary intended use of BLAST is for annotation of translation errors.

Evaluation of BLAST mainly focuses on identification of usability issues, understandability and proposal of redesign to overcome issues of usability. By allowing the subjects to explore BLAST, the usage and performance of the tool are observed and later explained.

In this usability study, five participants were involved and they were requested to perform user tasks designed to evaluate the usability of tool. Based on the user tasks required data is collected. Data collection methodology included interviews, observation and questionnaire. Collected data were analyzed both using quantitative and qualitative approaches.

The Participant’s technical knowledge and interest to experiment new interface shows the impact on the evaluation of the tool. The problems faced by individuals while evaluating was found and solutions to overcome those problems were learned.

So finally a redesign proposal for BLAST was an approach to overcome the problems. I proposed few designs addressing the issues found in designing the interface. Designs can be adapted to the existing system or can be implemented new. There is also a chance of doing an evaluation study on interface designs proposed.

Key words:

Usability, Evaluation, Annotation, Error analysis, Machine translation
# Table of Contents

1 Introduction .................................................................................................................. 10
   1.1 Goal ......................................................................................................................... 10
   1.2 Problem Formulation ............................................................................................. 10

2 Theoretical Background ............................................................................................... 11
   2.1 BLAST Error Annotation tool ................................................................................. 11
   2.2 Usability .................................................................................................................. 12
   2.3 Evaluating Usability ............................................................................................... 13
      2.3.1 User-based Methods ......................................................................................... 14
      2.3.2 Expert-based Methods ...................................................................................... 15
      2.3.3 Model-based Methods ...................................................................................... 15
      2.3.4 Comparisons of Methods ................................................................................ 16
   2.4 Data Collection ....................................................................................................... 17
      2.4.1 Questionnaire .................................................................................................... 17
      2.4.2 Interviews .......................................................................................................... 17
   2.5 Measuring Usability ............................................................................................... 18
      2.5.1 Performance Metrics ......................................................................................... 19
      2.5.2 Self-reported Metrics ......................................................................................... 20
   2.6 Prototypes ............................................................................................................... 22
   2.7 System / Product Evaluation in real time ................................................................. 24

3 Method .......................................................................................................................... 25
   3.1 Experimental Setup ................................................................................................. 25
      3.1.1 Machine Translation System .......................................................................... 25
      3.1.2 Error Typology .................................................................................................. 26
      3.1.3 Experimental Texts .......................................................................................... 26
      3.1.4 Gold Standard Annotations ............................................................................. 27
   3.2 Participants .............................................................................................................. 27
   3.3 User Tasks ............................................................................................................... 28
   3.4 Test Procedure ........................................................................................................ 29
      3.4.1 Pilot Test ............................................................................................................. 29
3.4.2 Real Test .......................................................... 33
3.5 Data Collection .......................................................... 33
  3.5.1 Questionnaire .................................................. 33
  3.5.2 Interviews ....................................................... 34
3.6 Prototyping .............................................................. 34
3.7 Decisions taken .......................................................... 35
4 Results .................................................................. 37
  4.1 Quantitative Results .................................................. 37
    4.1.1 Task Success ................................................... 37
    4.1.2 Task Success Rate ............................................ 37
    4.1.3 Time on Task ................................................... 39
    4.1.4 System Usability Scale ....................................... 40
    4.1.5 Expectation Measure ......................................... 40
    4.1.6 User’s Task Experience ...................................... 41
    4.1.7 User’s Overall Experience .................................. 47
  4.2 Qualitative Results .................................................. 48
    4.2.1 Open-ended Questionnaire .................................. 48
    4.2.2 Post-interview Results ....................................... 49
5 Analysis .................................................................. 51
  5.1 Observations ........................................................... 51
  5.2 Issues and Recommendations .................................... 54
6 Redesign Proposals ..................................................... 57
  6.1.1 Proposal of Designs ............................................. 59
  6.1.2 Modifications ..................................................... 67
  6.1.3 Follow up interview on redesigns ............................ 68
7 Conclusion ................................................................ 69
8 References ................................................................ 70
Appendices .......................................................................................................................... 72
Appendix A: Test Introduction paper .................................................................................. 72
Appendix B: Demographic Questions .................................................................................. 73
Appendix C: Experiment Tasks .......................................................................................... 74
Appendix D: The System Usability Scale ........................................................................... 77
Appendix E: The BLAST Usability Form ............................................................................. 78
Appendix F: Open-Ended questionnaire ............................................................................. 79
Appendix G: Interview questionnaire .................................................................................. 80
List of Tables

Table 1: Relative advantages and disadvantages of each usability evaluation method. Adapted from Dillon (2001). ................................................................. 16
Table 2: Tasks to be performed on BLAST ................................................................. 29
Table 3: Expectation measures of tasks .................................................................... 40
Table 4: Readings for overall tool’s usability ............................................................... 47

List of Figures

Figure 1: Graphical User Interface of BLAST ............................................................ 11
Figure 2: Classification of Errors. Fig ref to Vilar et.al (2006) ................................ 26
Figure 3: Classification of translation errors (multi-level) ....................................... 31
Figure 4: Classification of translation errors (2-level) ............................................. 32
Figure 5: Task Success rate for tasks 2,3,4 ............................................................... 37
Figure 6: Comparison between original files and annotated files for task 3 .......... 38
Figure 7: Comparison between original files and annotated files for task 4 .......... 39
Figure 8: SUS scores for every user ........................................................................ 40
Figure 9: Average expectation rating using Scatter plot. ....................................... 41
Figure 10: Usability measures for user 1 ................................................................. 42
Figure 11: Usability measures for user 2 ................................................................. 43
Figure 12: Usability measures for user 3 ................................................................. 44
Figure 13: Usability measures for user 4 ................................................................. 45
Figure 14: Usability measures for user 5 ................................................................. 46
Figure 15: Visualizing data using RADAR chart ..................................................... 47
Figure 16: Graphical User Interface of BLAST (Annotation mode) ..................... 57
Figure 17: Graphical User Interface of BLAST (Edit mode) ............................... 58
Figure 18: Graphical User Interface of BLAST (Search mode) ......................... 58
Figure 19: Proposal 1 (annotation mode) .............................................................. 59
Figure 20: Proposal 1 (edit mode) ........................................................................ 60
Figure 21: Proposal 1 (search mode) .................................................................... 61
Figure 22: Redesign proposal 2 ............................................................................ 62
Figure 23: Redesign proposal 3 (annotation mode) ............................................... 63
Figure 24: Redesign proposal 3 (edit mode) ........................................................... 64
Figure 25: Redesign proposal 3 (search mode A) .................................................. 65
Figure 26: Redesign proposal 3 (search mode B) .................................................. 66
1 Introduction

The main purpose of this thesis is to evaluate the user interface of a graphical annotation tool, developed by Sara Stymne. The goal of BLAST is to support human error analysis of machine translation output (Stymne, 2011).

To evaluate the user interface I was provided with a working software version of BLAST. Using the initial version of BLAST, usability tests were carried on. There are various methods and techniques to carry evaluation process. The evaluation process of BLAST was carried out based on an extension of the user-based evaluation approach.

The evaluation was formative, as the name suggests it was mainly focused on identifying problems and suggesting improvements to the design. The evaluation aims at the usage of the tool and the issues found by the users. The evaluation also examines whether the users could perform the tasks on time or not.

1.1 Goal

The goal of the study was to

1. Know user satisfaction
2. Identify usability problems
3. Propose redesigns to overcome issues.

Results are taken into consideration for improving the design.

1.2 Problem Formulation

The study was mainly aimed to answer the following questions regarding the annotations in BLAST.

1. What usability issues can be identified in the BLAST GUI?
2. What problems do users feel during annotations?
3. How satisfied are the users with the tool?
4. How can the tool be redesigned to minimize above issues?
2 Theoretical Background

2.1 BLAST Error Annotation tool

BLAST is an open source tool for error analysis of machine translation output (Stymne, 2011). It has a flexible graphical user interface and can be used with any machine translation system (MTS). Error analysis is the identification and classification of errors in a machine translated text. BLAST follows typologies and it works in three modes for adding new annotations, for editing existing annotations, and for searching among annotations.

Error annotation is a note or a comment made on some section of a sentence. The purpose of the annotations is to support applications, which produce some kind of useful results by analyzing the texts. BLAST handles error annotations and support annotation. Error annotations are used to annotate errors in Machine Translation (MT) output. The users of BLAST add these. Support annotations are used as a support to the user, which marks similarities into the system and reference sentences. The support annotations are normally created automatically by BLAST.

![Figure 1: Graphical User Interface of BLAST](image-url)
Figure 1 shows a screenshot of BLAST. The interface of the tool was observed in two parts, upper part shows the MT output to the user operating the tool and this area displays three sentences; source sentence, reference sentence and system sentence. The lower part of the tool displays the error typology, options for creating, updating annotations and navigation between the controls. Error typology follows a menu structure and user can activate the submenus by clicking on them.

The design idea of BLAST is to be flexible and allow full freedom for the user to use. BLAST provides compatibility with any error typology and possibility to mark errors in a sentence. User can view the automatically highlighted similarities between system and reference sentences. BLAST provides a search function for errors also.

One can say that BLAST is a well designed and properly developed tool by looking at its features, but one cannot say that this tool is usable and user satisfactory because BLAST has not been evaluated in terms of usability.

2.2 Usability

The concept of usability has been defined and explained by many people in many ways. Usability is often described as how well a system can be used or considered as the ability of the user to carry out the task successfully.

The International Organization for Standardization defines usability of a product as “the extent to which the product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.” (Alshamari & Mayhew, 2009)

According to (Alshamari & Mayhew, 2009) Usability is about

- Effectiveness - can users complete tasks, achieve goals with the product, i.e. do what they want to do?
- Efficiency - how much effort do users require doing this? (Often measured in time)
- Satisfaction – what do users think about the product’s ease of use?

which is all affected by

- Users – who is using the product/system?
- Goals – what actually are the users trying to do with the system?
- Context of use – how and where is the system being used?
Usability generally refers to the quality of being able to provide good service. It can also refer to making a product easier to use by matching its features with the user needs and requirements. Usability is a quality attribute that assesses how easy user interfaces or systems are to use.

Nielsen (1994) describes that usability can be defined by five quality components

- **Learnability**: How easy is it for users to accomplish basic tasks when they encounter the design?
- **Efficiency**: Once users have learned the design, how quickly can they perform tasks?
- **Memorability**: When users return to the design after a period of not using it, how easily can they reestablish proficiency?
- **Errors**: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- **Satisfaction**: How pleasant is it to use the design?

Depending on the type of application, each component might be more critical than another.

A user is involved in performing some act with the product or a system. For a broader view on usability, including the entire user experience: the user’s ability to use the system successfully and his/her thoughts, feelings and perceptions while using the system are coincidentally important (Tullis & Albert, 2008).

### 2.3 Evaluating Usability

Usability evaluation is an important activity in designing the user interface for various applications, which further leads to important discoveries about usability and creates opportunities for valuable design improvements. Usability evaluation refers to a set of methods through which evaluators examine usability related aspects of an application and provide judgments based on their human factors expertise. There are multiple ways to evaluate usability of a product depending on available resources, evaluator experience, ability and preference, and the stage of development of the product under review. According to Scholtz (2003), the three most discussed evaluation methods are:

- **User-based**: where a sample of the intended users tries to use the application
- **Expert-based**: where a usability expert makes an assessment of the application
- **Model-based**: where an expert employs formal methods to predict one or more criteria of user performance
2.3.1 User-based Methods

Testing an application with a group of users performing a set of pre-determined tasks are generally considered to produce the most reliable and valid estimate of a product’s usability. The aim of such evaluation is to examine the extent to which the application supports the intended users in their work. Based on Scholtz (2003), the user-based evaluations can be carried out in two ways.

- Formative Evaluation
- Summative Evaluation

2.3.1.1 Formative Evaluation

Formative evaluation is used to obtain user feedback on the software product design. Formative methods are informal; moreover, the goal of this method is to collect information regarding design and usability measures.

According to Bevan & Singhal (2009), formative evaluation helps to "form" the design for a product or service, which involves evaluating the product or a service during development, with the goal of detecting and eliminating usability problems iteratively. Furthermore, depending upon the design issues under evaluation, usability testing can be conducted using simple, low fidelity paper or foam mock-ups or higher fidelity software or presentation prototypes.

For usability testing to be an effective tool for understanding user interface design strengths and weaknesses, it needs to engage actual users in performing real work. Formative usability testing is the most reliable way to develop a truly usable product.

Formative evaluation involves many different tasks

- Identify and evaluate the goals
- Contributing to methodological choice
- Making valuable assessment
- Generating findings

2.3.1.2 Summative Evaluation

Summative evaluation is the formal way of evaluating. These methods are used to document and record the usability characteristics of a software product that involves a number of users.

In a typical user-based evaluation, test subjects (users) are requested to perform a set of tasks with the product. Depending on the evaluator’s primary goal, the user’s success at completing the tasks and their level of performance will be recorded. After the
completion of tasks, users are questioned to provide information on likes and dislikes through an interview. In this way, measures of effectiveness, efficiency and satisfaction can be derived. The main problems are identified and re-design advice can be determined.

Some user-based tests are unstructured, which involves the user and the evaluator interaction with the system jointly to gain agreement on what works and what is problematic with the design. Such participative approaches can be very useful for exploring interface options.

2.3.2 Expert-based Methods
An expert-based method refers to the usability evaluation, which is carried out by an expert, examining the application and estimating its usability. Users are not employed in such cases and the basis for the evaluation lies in the interpretation and judgment of the evaluator. Expert-based methods are encouraged since it can produce results faster and presumably cheaper than user-based tests.

The two common expert-based usability evaluation methods are Heuristic evaluation (Nielsen, 1994) and Cognitive Walk-through (Wharton, 1994). These two methods aim to provide evaluator with a structured method for examining and reporting problems with an interface. The Heuristic method provides a list of design guidelines, which the evaluator uses to examine the interface. The evaluator reports violations of the guidelines as likely, user problems (Nielsen, 1994). In the Cognitive Walk-through method, the evaluator first determines the exact sequence of correct task performance and then estimates the success or failure of the user in performing such a sequence. The Cognitive Walk-through method concentrates more on the difficulties users may experience in learning to operate an application to perform a given task (Wharton, 1994). In practice, usability evaluators tend to adapt and modify methods that suit the purpose and furthermore, experts employ a hybrid form of the evaluation methods.

2.3.3 Model-based Methods
Model–based approach uses a model of the usability evaluation situation to represent the interface design and produce predicted measurements of usability (Gray et al, 1992). These approaches to usability evaluation are least practiced.

Model-based evaluations like empirical evaluations are appropriate for identifying usability problems in quantitative approach. It uses cognitive and design models to evaluate user interfaces. In fact, it refers to the process of using a model of how the users would use a proposed system to obtain predicted usability measures by calculation. Model-based evaluations are rarely used in the evaluation of system
usability, since they are still limited and immature, expensive to apply and also there is limited guidance on how to apply them (Card et al, 1983). Moreover, model based evaluation techniques cannot be used to evaluate how system will be used in a real world context. The model-based approaches are regarded as limited and expensive to apply and their use is largely restricted to research teams.

2.3.4 Comparisons of Methods

The relative advantages and disadvantages of each method are summarized in Table 1. Since usability evaluators are trying to estimate the extent to which real users can employ an application effectively, efficiently and satisfactorily, the executed user-based methods always give the most accurate estimate. However, the usability evaluator does not have the necessary resources to perform such evaluations so, other methods must be employed.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>ADVANTAGES</th>
<th>DISADVANTAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-based</td>
<td>More realistic estimates of usability,</td>
<td>Time consuming, costly for a large sample of users,</td>
</tr>
<tr>
<td></td>
<td>Can give a clear record of important problems</td>
<td>requires prototype to occur</td>
</tr>
<tr>
<td>Expert-based</td>
<td>Cheap,</td>
<td>Expert-variability unduly affects outcome, may</td>
</tr>
<tr>
<td></td>
<td>Fast</td>
<td>overestimate the true number of problems</td>
</tr>
<tr>
<td>Model-based</td>
<td>Provides rigorous estimate of usability criterion, can be performed on interface specification</td>
<td>Measures only one component of usability, limited task applicability</td>
</tr>
</tbody>
</table>

Table 1: Relative advantages and disadvantages of each usability evaluation method. Adapted from Dillon (2001).

John and Marks (1997) compared multiple evaluation methods and concluded that no method is best and all evaluation methods are of limited value. It is generally recognized that expert based evaluations employs the heuristic method, which identifies more problems than other methods, including user-based tests. This may suggest that heuristic approach solves the problem including interface attributes that users do not experience.

Multiple expert evaluations produce better results than single expert evaluations. There are good reasons for thinking that the best approach to evaluating usability is to
combine methods e.g., using the expert-based approach to identify problems and inform the design of a user-based test scenario, since the overlap between the outputs of these methods is only partial, and a user-based test normally cannot cover as much of the interface as an expert-based method. The point of usability evaluation is to create products, which can provide ease of use and minimize the risk of errors, especially critical ones.

2.4 Data Collection

There are two ways of collecting data, qualitative and quantitative. Quantitative methods focus on numerical values rather than on meaning and experience. Quantitative methods like experiments, questionnaires and psychometric tests provide information, which was easy to analyze statistically and fairly reliable. The qualitative data are collected through interviews and user tests. Quantitative method is a scientific and experimental approach and is criticized for not providing an in depth description. Qualitative methods provide a more in depth, value and a trustworthy description. Generally, usability studies are closely related to traditional psychological studies, which mainly focus on quantitative measures. These measures can sometimes be augmented with some qualitative aspects.

2.4.1 Questionnaire

The questionnaire is a form of query research in which the main idea is to find out how and what do the participants think about the interface? This can be known by simply asking them in person. However a questionnaire is easier for the evaluator to analyze data after evaluation.

Different types of questions can be included in the questionnaire in a user study. Questionnaire includes general demographic questions about the participant, open-ended questions and scalar questions. Open-ended questions provide space for the participants to express their feeling about the interface or a system. Scalar questions allow participants to rate a specific statement on some kind of numeric scale.

2.4.2 Interviews

In any evaluation, interviews are conducted for several reasons and these interviews are carried out by different approaches. Interviews are categorized into four types. They can be unstructured (informal), structured, semi-structured or group interviews. An unstructured interview allows the evaluator to ask/interview participants in a way such that participants open up and express their ideas and feelings. These interviews give data in huge amount, which sometimes makes it difficult for the evaluator to analyze the data in a proper way. Informal interviews are done spontaneously. Informal
interviews are like a general conversation between evaluator and participant, which may lead to time consuming discussion.

The second type of interview is structured interviews. These structured interviews are easier to replicate when compared with an unstructured interview. Here evaluator is provided with a proper question and participants are provided with a choice to choose from a set of answers. These interviews are really useful when the evaluation or an interview is carried out for a specific purpose. This is in contrast to the questions in unstructured interviews, which may be predetermined but the participant is allowed to answer freely with choice and freedom like in an unstructured interview.

A semi-structured interview is a blend of structured and an unstructured interview. It has its pros and cons depending on the situation. A group interview is usually referred to a focus group, which makes them to have an open discussion on a predefined project, which was more structured. The main purpose of these interviews is to allow people to relate and interact with the each other in a realistic social setting. Sometimes this will give opposite effects, as participants are not willing to share their feelings.

The most important part of the evaluation is to decide upon which interview method to be followed and which questions to be asked to participants. The evaluator must be conscious about how to ask a question and how it is interpreted. In semi-structured interviews the theme of the answers is usually defined by the questions asked. This simplifies the analysis process for an evaluator.

2.5 Measuring Usability

There are several methods and techniques that can be used to reach more informed design decisions. Introspective methods such as heuristic evaluation or cognitive walk-through employ usability experts to evaluate a product, by checking against design principles or stepping through tasks simulating a user. Usability testing is a method to evaluate the product by testing it directly on users.

One popular usability test technique is Think-aloud, where the user is asked to continuously say his/her thoughts out loud while carrying out tasks in the system. Think-aloud require participants to tell evaluator what they are thinking and doing while performing a task. The participants are usually instructed to keep thinking aloud, acting as if they are alone in the room speaking to themselves. Think-aloud protocols are tape- and/or video-recorded and then transcribed for content analysis.

Usability metrics are the measurements used to collect data in usability tests. The choice of metrics should be based on student goals, business goals, user goals, budget, time
and available technology to collect and analyze the data. Choice of metrics depends upon the type of usability study. Study on usability could be formative or summative. It is very important to establish a set of baseline while evaluating new products than performed evaluation iterations. Usability metrics should possess basic properties like they must be observable, quantifiable or possible to count in some way. The classification of usability metrics is by the aspects of effectiveness, efficiency and satisfaction. According to Tullis & Albert (2008) usability metrics can be divided into performance, issue-based and self-reported. Metrics from these categories can additionally be combined or compared.

2.5.1 Performance Metrics

Every type of user behavior can be measured in one way or the other. All performance metrics are calculated based on specific user behavior. These metrics rely not only on the user behavior but also on the intended use of tasks. Without having specific tasks, performance metrics cannot be collected. One cannot measure the success if the user is aimlessly using the tool. Performance metrics are the best way of knowing how well users are actually using the product. This is the best way to evaluate effectiveness, efficiency and satisfaction of any product. Efficiency metrics are generally used to measure how much cognitive or physical effort was needed by the user in order to complete the task.

According to Tullis & Albert (2008) the basic types of performance metrics are the following:

• **“Task Success”** is a very common way to measure effectiveness. It tells us whether the users are able to complete the task or not during evaluation. Task success can be measured in binary format (1= Success and 0= Failure). A stopping rule can be employed to limit the user how long he/she takes to complete the task successfully.

• **“Task Success rate”** is another way of calculating effectiveness. It tells us the actual success rate of the tasks done. The actual success rate can be measured finding successful tasks from the total number of tasks done. Tasks are said to be successful by comparing with certain standard or based on a specific rule.

In this study, by comparing the annotations made by participants to gold standard annotations task success rate can be measured. The files which have annotations exactly same as the gold standard annotations was treated as success. The 10 sentences from participant were compared to the gold standard’s
10 sentences. By comparing both the sentences, sentences with similar annotations were considered as the most successful out of the total sentences and rate was calculated. The process continues for all the participants to calculate the task success rate for the tasks that they had performed.

- **“Time on task”** also called, as completion time is an efficiency metric that tells how many time units the task takes to complete. This is usually measured in seconds or minutes. Defining the exact start and stop points is really important to measure time.

### 2.5.2 Self-reported Metrics

Performance metrics give only “what but not the why”. To know why the problem does exist, we need to observe the data. This process of data collection by observations can be called as self-reported metrics, which gives a better understanding of why the problems arise and how they can be fixed. The most common use of self-reported metrics is to know an overall measure of perceived usability that users are requested to give after interaction with the system or product. According to Tullis & Albert (2008) the best ways to collect self-reported metrics are

- **Post - Task ratings**
  These are used to give insights into the satisfaction and frustrations on specific tasks performed. These ratings have been collected just after every single task is finished. Participants are requested to rate the difficulty of the task performed. To rate difficulty, interest and learnability a scale was provided along with the experiment task paper. According to rating scale 1=very easy and 5=very hard (See in Appendix E).

  - **“Learnability”** is the extent to which something can be learned. Learnability measures how easy and quickly the user become proficient with using the tool or product. Learnability can be measured using almost any performance metric (time-on-task, task success) over time. Learnability can be measured by collecting any metric over several points in time. This is impractical since the participants are not available over extended periods of time.

  - **“Difficulty”** measure is used to know up to what extent the difficulties are about. Difficulty measures how easy or hard the user felt while performing tasks and using the tool on a whole. Difficulty in other terms can be called as the ease of use. Difficulty can be measured on individual
task and on the whole set of tasks. The results can be used to know the level of difficulties and can be represented in radar graph. Radar graph is a graphical method of displaying multivariate data in the form of a two-dimensional chart of three or more quantitative variables represented on axes starting from the same point.

- **“Interest”** measures how interesting or boring the user felt while performing tasks and using the tool on a whole. Interest can be measured on individual task and on the whole evaluation. The results can be used to know the level of difficulties and can be represented in radar graphs when compared with difficulty, learnability and satisfaction.

**Post - Study ratings**

These are used to give an overall measure of the user experience after they had completed their interaction. These ratings are collected after the evaluation process is done. The result reflects the whole evaluation. This ratings can be obtained by an in-depth reading and an open-ended questionnaire. It is achieved by calculating Expectation Measure and SUS (System Usability Scale). These two scores can be displayed as a scatter plot.

- **“Expectation measure”** is used to compare how easy or hard the participant thought the study would be? A different approach was proposed by Albert and Dixon (2003) to assess subjective reactions after every task. Participants are expected some tasks to be easier than others. Participants were asked to rate how easy/hard the task actually was. The “before” rating is called expectation rating and “after” rating is called experience rating. A five point rating scale is used to take both the ratings. According to rating scale 1=very easy and 5=very hard. Based on the rating given by the participants an average expectation rating and an average experience will be calculated and later can be displayed as a scatter plot.

- The **“System Usability Scale (SUS)”** is a very commonly used usability questionnaire that provides a single reference score for participants’ view of a product’s usability. According to Brooke (1996), the score is calculated based on the answers of ten statements, each rated on a 5-point scale ranging from 0 to 100, where 100 means perfect usability. To calculate the
SUS score, a formula is used. The formula is SUS score = ratings * (ratings - 1). 2.5 to obtain the overall SUS score should multiply the added score. SUS is technology agnostic, relatively quick and easy to use and the provided score is easily understood by people who have little or no experience in usability (Bangor, 2008). In a study done by Tullis & Albert (2008), SUS yielded the most consistent ratings at relatively small sample sizes in a comparison between several questionnaires for post-study ratings. SUS uses both negative and positive statements to keep the participants alert. It is important to analyze only the single score but not the individual statements.

2.6 Prototypes

A prototype is a draft or an initial version of a system. It is not intended mainly for real use, but it is used to conduct experiments to resolve different issues. Prototypes allow user to explore new ideas before investing time and money into development. A prototype can be anything from a paper drawing, click-through of a few images or pages, or a fully functioning interface.

A prototype of a user interface mainly consists of screens with data fields, menu, function keys, buttons etc. Screens can be drawn in several ways according to the users' willingness. According to the Lauesen (2005) there are four most used types of drawing prototypes.

- **Hand-drawn mock-up screens**
  By using paper and pencil the designer will draw the screens. User changes screen after screen by putting or removing papers on a flat surface like a table during usability evaluation. Data can be entered by writing into the data fields using pencil by the designer.

- **Tool-drawn mock-up screens**
  The designer draws the screens graphically using a tool on a computer. Microsoft Access, Visual Basic and Caretta GUI design studio are some examples of such tools, which allow the designer to draw screens. The designer uses these screens as the user's hand-drawn screens. These screens look more real than the hand-drawn screens.

- **Screen prototype**
  These screens are shown on the real computer. These screens are drawn and demonstrated with little functionality. User may enter data into the fields or
press some buttons but it does not directly affect the prototype unless these have some functionality integrated with the system. In general user needs to draw several screens for several versions of the system to reflect working nature of a real time system.

- **Functional prototype**
  These prototypes look similar like screen prototypes with buttons, menu points with working functionality. These prototypes can bring to pass information from one screen to another, while navigating. These screens can also fetch data from databases when connected with functionality.

These four kinds of prototypes can be combined and used according to the user requirements. All four kinds of prototypes can detect usability problems. They are good enough for defining what to develop, and for discussing with users and customers. Although the prototypes can reveal the same problems, a problem that looked like a task failure with the mock-up screen may become a minor problem with a functional prototype. The main difference between the prototypes is the time they take to develop. If it takes more time, designers are not interested to change the prototype radically.

**Advantages of Prototypes**

- Gives end users an idea of how the system looks like
- Provides user quantifiable feedback to designers/developers
- Provides value for designing end user interface
- Since users know what to expect, it facilitates system implementation
- Provides better results in high user satisfaction
- Provides enough exposure to designers for developing more reliable system
- Technical features can be well tested
- Helps to find potential risks

**Disadvantages of prototypes**

- Leads to insufficient analysis
- Not suitable for large applications
- Rapid prototyping may gloss over essential steps in system development
- Lacking flexibility
- Produces an inadequate system for organizational needs
2.7 System / Product Evaluation in real time

There are two types of evaluation, which are highly followed and practiced. They are Formative and Summative. Formative evaluation affects the object of evaluation and is often conducted during the product/system development. Summative evaluation is conducted after the product/system is developed. By considering a technical example of usability studies, a formative study can be part of an iterative design process, where the results lead to redesign. A summative evaluation is the assessment done where; the end users have the product with them and tell ideas and opinions about the usability of a product, which does not disturb the product implementation. Depending on the situation of the evaluation, the role of the evaluator varies. The evaluator is expected to give objective input on whether the product or system development should continue or to be terminated in a summative evaluation. In formative evaluation the evaluator has to work closely with the people responsible for the product/system development.

To carry out any kind of evaluation, the following usability criteria were suggested.

1) How difficult is it to use?
2) How long does it take to learn in order to use it?
3) How often do errors occur and how serious are they?
4) How much mental stress does the user undergo?

Coming to errors it is always considered in two ways

1) Number of errors in the form of breakdown of system
2) Errors generated by the users generally

An evaluator may use any kind evaluation study. Evaluation study could be formative or summative, but the study should be answering the above questions in one or the other way. In this evaluation, I framed a questionnaire to design a well-thought-out usability study. Questions followed were

- What type of participants do I need?
- How many participants do I need?
- What kind of tasks to be performed?
- How do I compare data from every single user to other?
- Do I need to adjust order of tasks?

I tried to highlight the main design problems that might occur while using this system. Selecting participants was another task for me to focus on.
3 Method

This section gives a complete overview of how the evaluation process was carried out and also the methods and techniques followed to evaluate the BLAST.

Within the area of research on graphical user interfaces for Machine Translation System, I consider that the design and development of a suitable usability evaluation methodology should enable the quantification of relevant usability attributes and especially, identification of BLAST weaknesses or issues.

This methodology was inspired from previous studies based on evaluating graphical user interfaces (GUI). This methodology was used to conduct usability evaluation of the BLAST tool, which consists of user testing during users' walk-through along the BLAST interface, guided by a set of predefined steps. This approach was chosen by having in mind, that usability could only be measured during task performance. The evaluation methodology was based on criteria expressed in terms of objective performance measures in systems use, as well as in terms of user’s subjective assessment.

An appropriate evaluation methodology was essential, which will enable system validation on one hand and provide a means to compare achieved results with those obtained by previous system evaluations on the other hand. This same usability evaluation methodology could be applied in evaluating the redesigned BLAST version.

3.1 Experimental Setup

A user study was performed where the evaluator took the feedback of the participants through several measures while the participants were requested to perform several tasks on BLAST. The user study was mainly related to error annotation on sentences, which were translated from English to Swedish. The experiment was set up in a way that participants have to read and annotate the generated translation text from English to Swedish.

3.1.1 Machine Translation System

Output from an English – Swedish Machine Translation System was used in this evaluation study. It was standard phrase-based statistical machine translation system, which had been built using the Moses toolkit (Koehn et al., 2007). (The System evaluated was the baseline system from Stymne and Holmquist, 2008) and was trained on 701157 sentences from Europarl (Koehn, 2005).
3.1.2 Error Typology

The Error typology used in this evaluation was mainly adapted from Vilar et al. (2006). The Errors were mainly classified into the five base categories of Vilar et al (2006): missing words, word order, incorrect words, unknown words and punctuation as shown in figure 2.

![Figure 2: Classification of Errors. Fig ref to Vilar et.al (2006)](image)

3.1.3 Experimental Texts

The texts that were used in this test were collected from Europarl (Koehn, 2005). The Europarl data source text originally was a huge collection of sentences from the European Parliament proceedings. 10 sentences were considered for the evaluation study for each task. The sentences selected were not related to each other. There were total 40 sentences for 4 tasks and the sentences used for one task were not repeated and different from each other. For task 2, sentences used were of plain text where the participant was allowed to annotate on the sentences freely. For task 3, sentences used were of partially annotated text; the user can do annotations or remove annotations on these 10 sentences. For task 4, sentences used were already annotated with different kind of errors where participants need to find and correct the annotations. Later in task 5 for doing a search and count for specific kind of errors, another 10 sentences were selected by having a particular type of error.
3.1.4 Gold Standard Annotations

An experienced expert annotator who is a native speaker of Swedish annotated the sentences that were selected to use in the evaluation. The annotator has a good knowledge of MT and linguistics. These annotations were called as gold standard annotations and were used for comparing the annotations made by participants. By comparing the annotations made by participants to gold standard annotations, task success rate can be calculated. The sentences, which have annotations of same similarity level as the gold standard annotations, were treated as success. By comparing both the annotations made by participants and the gold standard annotations, sentences with similar annotations were considered as the most successful out of the total sentences and task success rate was calculated. The process continues for all the participants to calculate the task success rate for the tasks that they had performed.

3.2 Participants

This evaluation mainly deals with the content, quality and relevance of a GUI. I decided to execute this usability study both in qualitative and quantitative way with five users.

Actually there was no such fixed number of users that will always be the right number to perform the evaluation. But testing with five users could be helpful in discovering problems in an interface, given some conditions. That number allows for a range of answers and also gives the evaluator the ability to go into more detail with each user than if there were, say, double or triple that number involved.

In finding the participants to carry out the evaluation, several people were contacted for the usability study; three participants who participated during the pilot test procedure were different from the five participants who participated in the main evaluation procedure. All the participants were students from Linkoping University who study Computer Science.

• Participants for the pilot test procedure

The main aim of engaging participants of test procedure was to figure out if the working nature of the BLAST and to calculate the average time for performing and completing the task. A look alike test of the main procedure was conducted and participants were asked to follow the instructions and continue with the evaluation. Three users carried out the pilot test procedure. The users were aged between 21-25 and they were interested to learn what they were actually doing. They were capable of understanding the given instructions.
• Participants in the main evaluation

The total number of five users were contacted and appointed to carry the evaluation process. None of the participants were usability experts; however two of them had previous experience in evaluating user interfaces. The participants were aged between 18-25 years. The participants were interested in listening information about the tool, they were experienced in using software applications and capable of understanding and performing the task given. All the users were native Swedish speakers having a good knowledge of English, since all test instructions were given in English.

Recruitment of participants for pilot test and real test was not an easy job. Finding the participants who were truly interested in evaluation studies and spending time in evaluating the GUI of BLAST took a long time. Convincing participants to participate and explaining them how, when and where to perform the tasks was a challenge.

3.3 User Tasks

The experimenter (observer) clarified the nature of the task, and explained how the initial interface works. During this session, the users were oriented towards evaluating every feature of the interface design that they think should have changed, based on their own experience. They were not obliged to follow the guidelines, but they were asked to speak aloud and justify every decision that they made, without necessarily showing the specific guideline and instructions that they were following. The users had one hour to check and write the feedback about the main features of the initial version of the interface.

The full task instructions are given in English. Each instruction paper began with a brief scenario explaining the goal of the task from a user’s perspective, followed by task instructions and instructions on how to report the task as completed. Mostly, the participant was asked to verbally report when they thought that they were done; reporting instructions that differ from this are noted separately. The reports were a part of the success criteria for each task. A summary of the user tasks was presented in the table 2 (see Appendix C).
<table>
<thead>
<tr>
<th>Task 1</th>
<th>Try to upload and run a given BLAST file.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 2</td>
<td>Make annotations on the uploaded file.</td>
</tr>
<tr>
<td>Task 3</td>
<td>Make additional annotations in a given file (previously annotated).</td>
</tr>
<tr>
<td>Task 4</td>
<td>Find, remove and change errors in annotations in a given file (Previously annotated).</td>
</tr>
<tr>
<td>Task 5</td>
<td>Find out how many word order errors exist in a given file</td>
</tr>
</tbody>
</table>

Table 2: Tasks to be performed by subjects on BLAST

3.4 Test Procedure

As part of this evaluation, test procedure was carried out initially by a pilot test and a real test. The pilot test procedure was to ensure and declare time limits and other limitations of the real test. Later real test procedure was followed.

3.4.1 Pilot Test

The pilot test helped the evaluator to minimize the risk that the real test participants would face unexpected problems. The pilot study served as a feasibility test for this evaluation. Three participants participated in the pilot test. Participants were asked to say whether they are previously experienced in evaluating user interfaces. Participants were asked to read the instructions and fill in the user details. As per the evaluation users were requested to perform tasks one by one. Users were informed to feel free and use the tool without taking time into consideration. Users were requested to discuss what they felt while evaluating the tool for the first time. As an evaluator, my task was to note down the time taken for each task by each user. By calculating the average time of the readings from users for each task, time limits were set. Participants were requested to provide some feedback about the evaluation and the tool. One of the pilot test participants performed normal usability testing and other two used think-aloud testing technique. After the pilot test was done, my task was to analyze the whole testing procedure to identify any adverse effects caused by the procedure as a whole and the remedies to reduce them.

While performing pilot test the participants found it hard to annotate the errors according to the classification of translation errors shown in figure 3. The multilevel classifications of translation errors seemed hard to the participants, as they were not aware of the error types and their classification. Participants might have the knowledge of guessing the error type but they were not sure to annotate as a specific error. Participants raised the point of decreasing the multilevel classification to a 2- level
classification. Later discussing the issue with my supervisor we came to an agreement by changing the multilevel classifications to 2-level classifications which helps the participants who were not having the proper knowledge of linguistics to make quality annotations. This change of classifications provides ease of use and increase confidence in participants that they could do proper annotations. According to the agreement the system was changed to support 2-level error classification showed in figure 4 and used in real test.

Classification of translation errors (multi-level) can be seen figure 3. The classification shown in the figure 3 was incomplete as the multi-level error classification is too large to represent. This is just to give a basic idea to the reader that this classification was of multilevel and then it was reduced to a 2-level classification as shown in figure 4.
Figure 3: Classification of translation errors (multi-level)
Figure 4: Classification of translation errors (2-level)
3.4.2 Real Test

Participants were comfortably seated and provided with a working BLAST tool. I was sitting beside the participant, initially demonstrating the BLAST tool on a laptop with the working prototype. I gave a small demonstration of three minutes to each and every participant telling how and for what BLAST can be used. After the introduction, including information about the tool and their right to quit at any time, demographic data was collected. The participants were informed that they would be using a working prototype of BLAST. When the test began, participants were given instructions on a paper (refer to Appendix A). They were requested to report verbally when they assumed that they had finished the task or had decided to give up attempting. When they did any of these or reached the stop rule they were asked to fill in a BLAST usability form (see Appendix E). No task assistance was given during the test, except to explain the instructions if needed, and the participants were not informed that the tasks were timed, unless they asked. After all five tasks the evaluation was finished. When all the tasks were done the participant was requested to fill in the SUS form and the concluding open-ended questionnaire. This ended the test session. Before departing they were offered coffee or tea as thanks for their participation. The equipment used for the test was a laptop, optical mouse, papers and pens.

3.5 Data Collection

The user study, which was conducted for this thesis focused on both qualitative data and quantitative data. It was collected in the form of observations, open-ended questionnaire and interviews. While developing the task to evaluate it was always important to think about the participants who were going to use the system. The evaluation was mainly about knowing the user experience about the BLAST. Actually in order to perform tasks in reality by reading task instructions, the tasks were designed and developed by my supervisor and me. Task instructions were also developed and given by thinking in the participant’s point of view.

The data were analyzed qualitatively by identifying the level of user experience. Interviews and questionnaires were prepared targeting to find some usability issues. The data response, which was collected from participants, was relatively different from one another.

3.5.1 Questionnaire

Formulation of questions had been always important for an evaluator such that he should be able to provide participants with ambiguous free text.
The demographic questions and open-ended questionnaire can be found in Appendix B and F.

**Demographic Questions**

Demographic questions consisted of a set of questionnaire, which mainly questions participant’s gender, age, education and experience. Questions about Experience were mainly targeting the participant’s level of experience in reading, understanding of English and previous knowledge of usability evaluation. Data were collected by asking the participants to fill in the given Demographic questions form (see in Appendix B). The data from 5 users were collected and saved for an evaluation study. According to the data provided by the participants, two of the participants were under age 20 and remaining three participants were in 21 to 25 age group. 4 of 5 were male and one is female. Three of the participants answered with a high school degree and two answered with bachelor degree when asked about their high level of education. When asked about experience in evaluating interface 2 participants answered with yes and one of them did evaluation two times and the other did once.

**3.5.2 Interviews**

In this study I have chosen semi-structured interviews in which all participants were asked the same set of questions. In semi-structured interviews the theme of the answers was usually defined by the questions asked. This simplifies the analysis process for an evaluator who evaluates. The interview questions can be found in Appendix G.

**3.6 Prototyping**

Prototyping is the process of building an experimental system, which can be used for evaluation. Prototyping can be done quickly and inexpensively. When it comes to prototyping in this evaluation study, the main task was to provide designs to overcome the usability problems, which were found by the evaluator during the evaluation. Usability problems found by questionnaire and interviews were also considered.

To plan and develop a good redesign proposal there is a need to come up with multiple designs and select the best one. Then I planned to develop multiple designs for BLAST. So to develop such a good design, it was very important to make a good “Choice of Prototype”. Which kind of prototype should I use? Does the chosen prototype fulfill the requirements and bring up needed results? By taking these two questions into consideration, I decided to work with paper-based and tool-based prototypes.
• **Paper-based prototypes**

I developed a sketch for screens that show all necessary data for the user tasks. I made a paper and a pencil draft of the tool without showing many options. I created the paper sketches for three modes of operation of BLAST, i.e. annotation mode, edit mode and search mode. I tried to add some buttons by discussing with some of the participants what actually they felt was missing in the actually in the tool. After creating screens I took feedback from two participants by showing the navigation between the paper prototypes. Feedback was collected in a casual way by discussing what they felt good and improved compared to previous design they used. I made the required changes and some additions to the prototypes as for the suggestions provided by the participants. Later I started to work on tool-based prototypes.

• **Tool-based prototypes**

I used tool-based prototypes after drawing some conclusions from the paper-based prototypes. I choose Caretta GUI design studio professional as the drawing tool for developing a tool-based prototypes. The basic idea of using this tool was to develop prototypes in an attractive way such that the design represents the actual user interface. I tried to implement exactly what was done on paper-based prototypes. The developed prototypes were later shown to users and then opted for the redesign proposal.

3.7 **Decisions taken**

After discussing why, how and what about the BLAST’s usability method of evaluation, it is equally important to state the choices made by the evaluator before, during and after the evaluation. The ideas and choices were made by the evaluator and explained in detail.

**Choices:**

The choices made by evaluator in this evaluation were

**Choice 1: Reducing the annotation levels to 2-level annotations**

**Description:** As an evaluator, I wanted to reduce the multilevel typology order to a 2-level typology. This is due to the observations made by the evaluator during the pilot test. In pilot test the participants experienced problem to understand the topological order of errors and mark with the appropriate error type. User managed to mark till the
second level of typology, so the evaluator made a choice to reduce the level of annotation to 2-level annotations. This choice was made to make the participants properly mark the errors without any ambiguity in selecting subsets.

**Choice 2: Choosing 5 participants for evaluation**

**Description:** This was one of the choices made by the evaluator. There was always an argument between usability experts about carrying out evaluation by 5 participants. A group of the experts says 5 participants were not sufficient to evaluate a tool and the other group supports evaluation by 5 participants and they even say 5 members are sufficient to find more issues and problems during the evaluation. However after referring to Nielsen and Landauer (1993) and open discussions about this issue, it was my decision to go with 5 participants. One other reason for 5 users is also the lack of participants available due to the total evaluation time. In this evaluation the time taken was high compared to other usability evaluations.

**Choice 3: Metrics to measure usability**

**Description:** There are several ways to measure usability. Depending on the type and aim of evaluation certain choices needed to make. Some choices were taken by me in choosing the metrics suitable to collect and can be presented as results. There are several metrics mentioned in Tullis & Albert (2008), but the evaluator opted for performance metrics (task success, task success rate, time-on-task) and self-reported metrics (post task and post session ratings). I found that these two kinds of metrics could be suitable to evaluate the tool and represent its usability.

**Choice 4: Informal interviews and its questionnaire**

**Description:** Actually informal interviews weren’t the part of this evaluation study but after reading some successful evaluation stories from the Internet, the evaluator’s idea was to have a kind of interview, which was informal just after the evaluation, which opens up the participants to speak.

**Choice 5: Choice of prototyping**

**Description:** As mentioned in section 2.9 there are four types of drawing prototypes. I chose paper-based and tool-based prototypes to draw the prototypes of BLAST’s redesign. Evaluator chose to use these two types because these prototypes do not need to have any functionality included while presenting. As this evaluation proposes only redesign models for BLAST, I thought using paper-based prototypes first and then implementing in a graphical format would be appropriate and sufficient for this evaluation.
4 Results

This section analyzes the evaluation carried out by the evaluator and will report the results of the tasks, issues and overall experience of the tool. This section of report mainly describes the collected data as quantitative and qualitative results in two sections 4.1 and 4.2.

4.1 Quantitative Results

This section describes the results of performance metrics and self-reported metrics.

4.1.1 Task Success

Soon after performing each task, that task is given a “success=1” or a “failure=0”. Here the task is considered to be successful when the participant is able to finish the task by reading and understanding the task instructions. If the participant was not able to finish the task it was considered as a failure.

The percentage reaching 100% tells us that the overall task success if fulfilled and all tasks are performed. All the participants managed to perform and finish all the tasks within time irrespective of what they had done and regardless of what outcome it would result.

4.1.2 Task Success Rate

The participants were requested to perform the tasks as per the Experiment tasks (see Appendix C) and were asked to save the annotated file in specific form as mentioned test introduction paper. Files were saved under the user names of the participants and later compared to the gold standard annotations. The files were saved for tasks 2, 3, 4 mentioned in Appendix C.

![Figure 5: Task Success rate for tasks 2,3,4](image-url)
As shown in figure 5, on the horizontal axis we have the level of annotation and on the vertical axes we have percentages. Level-1 is represented for 1-level annotations; Level-2 is for 2-level annotations of error classification. Based on figures, data can be interpreted as follows

**Description:**

**Task 2:**

The gold standard annotations for task 2 when compared with raw annotation results as zero initially. But once after the participants annotating the file, those sentences can be compared to the gold standard annotations. After comparing the annotations of all participants the average of the results was collected. As per the averages collected the value for 1-level annotations is 0.3, which is relatively high compared to zero initially, and the 2-level annotations rate is 0.1. The value of 1-level annotations is high compared to 2-level annotations. This tells us that the participants managed to do annotations till 1-level and somehow failed to do 2-level annotations.

**Task 3:**

This task was to make annotations on a partially annotated file. Initially the file given to annotate was compared to the gold standard to calculate the difference. Then afterwards the files annotated were compared and the average was calculated. The results before the evaluation and after the evaluation were compared to both the levels. The results before and after the evaluation are shown in figure 6. The value before was 0.5 and the value after was 0.7 for 1-level annotations. This rating shows that the users performed better in 1-level annotations. The value before was 0.1 and the value after was 0.6 for 2-level annotations. There was also a gradual improvement on the original value to the annotated value.

![Figure 6: Comparison between original files and annotated files for task 3.](image-url)
Task 4:

This task was to find errors and correct them on a previously annotated file. Initially the file given to annotate was compared to the gold standard to calculate the difference. Then afterwards the files annotated were compared and the average was calculated. The results before the evaluation and after the evaluation were compared to both the levels (see figure 7). The value before was 0.6 and the value after was 0.4 for 1-level annotations. This rating shows that the user’s performance was bad in 1-level annotations. The value before was 0.6 and the value after was 0.2 for 2-level annotations. There was also gradual deterioration the original value to the annotated value. By this graph one can say that the user performance was really bad at both the levels.

![Figure 7: Comparison between original files and annotated files for task 4.](image)

Task 5:

In task 5, counting the number of word order errors in an annotated file is the question. There were 3 “word order errors” in a given file. The answer for this question is correctly answered by 3 participants with 3 and other two answered with 2.

4.1.3 Time on Task

The average time calculated from the pilot test was considered as the time limit and the participants were informed to finish the task in a prescribed time. The time limit for each task is 15 minutes. The participants were informed about them if they were about to reach the time limit. None of the participant crossed the time limit. All the participants completed all the tasks properly on time.
4.1.4 System Usability Scale

The ratings given in the SUS form by the participant were collected. Ratings should not to be considered individually but could be considered as a combined rating. It is more convenient to represent SUS scores as percentages as they are on a scale of 0 to 100. The overall SUS score is obtained as mentioned in section 2.4.1. Based on the rating given by the participants, the SUS scores were shown in figure 8.

![Figure 8: SUS scores for every user](image)

We know from the section 2.5.2 that the SUS consists of 10 statements, which were half positively worked, and half negatively worked. Based on the results provided by the participants, data can be represented as follows. User 1 thinks that BLAST is 52% usable and user 4 thinks that the system is 32.5% usable. The highest percentage of the usability is 52% and lowest is 32.5%. By calculating the difference between the highest and lowest scales, the usability difference was calculated. The difference was 19.5%.

4.1.5 Expectation Measure

The 5-point rating scales (1=very easy, 5=very hard) are used for taking both the ratings before and after the task. The quadrants for the scatter plot got changed compared to the original scatter plot according to Tullis & Albert (2008). For each task average expectation rating and an average experience rating are calculated. Calculated results are tabulated as shown in table 3.

<table>
<thead>
<tr>
<th>Task</th>
<th>Expectation</th>
<th>Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task1</td>
<td>2</td>
<td>2.2</td>
</tr>
<tr>
<td>Task2</td>
<td>2.6</td>
<td>2.6</td>
</tr>
<tr>
<td>Task3</td>
<td>2.2</td>
<td>2</td>
</tr>
<tr>
<td>Task4</td>
<td>2.4</td>
<td>2.4</td>
</tr>
<tr>
<td>Task5</td>
<td>1.8</td>
<td>3.8</td>
</tr>
</tbody>
</table>

Table 3: Expectation measures of tasks
Based on results obtained scatter plot is generated. The four quadrants of the scatter plot provide some interesting information about the tasks where and what to improve. In the lower right are the tasks that the participants thought would be hard but turned to be easy. This is called “promote it” quadrant. In the upper left are the tasks that the participants thought would be easy but turned to be hard. This is called “Fix it fast” quadrant. In the lower left are the tasks that the participants thought would be easy and turned to be easy. This is called “Don’t touch it” quadrant. In the upper right are the tasks that participants’ thought would be hard and were actually hard. These tasks might have the opportunity to make improvements. This is called “Big opportunity” quadrant.

Now by seeing the figure 9, we can interpret that for the tasks are in the lower left quadrant. This says that tasks are just working fine. This tells not to make any changes, which could have a negative impact. So there is no need to modify tasks 1, 2, 3, 4. But task 5 jumps in to fix it fast quadrant, which means the users were dissatisfied and evaluator should focus on, and need to fix it.

4.1.6 User’s Task Experience

This section explains the metrics collected from the participants in two ways, results on the individual task basis for every single user and results on the whole evaluation. Based on the evaluator’s convenience and ideas for evaluation, Only 4 tasks out of 5 were considered. Tasks were considered as task 2 = make annotations, task 3 = change annotations, task 4 = remove annotations and task 5 = search for annotations. Learnability, difficulty and interest were considered while generating graphs.
User1:

Figure 10 shows the results of learnability, difficulty and interest given user 1 for the tasks 2,3,4,5.

![Usability measures for user 1](image)

**Learnability:**

User rated by 2 for the tasks 2,3,4 and for the task 5 user rated with 3. For tasks 2,3,4 user learned a little based on the tasks, but there was an improvement for the task 5. After 3 tasks, user was a bit experienced and learnt using it.

**Difficulty:**

User rated by 2 for the task 2, rated by 3 for tasks 3,4 and for the task 4 users rated with 5. For task 2 users felt the task was a bit difficult, for tasks 3,4 user felt more difficult to perform, and then in task 5 users felt even more difficulty.

**Interest:**

User showed the same level of interest throughout the evaluation irrespective of the tasks.
User 2:

Figure 11 shows the results of learnability, difficulty and interest given user 1 for the tasks 2,3,4,5.

![Graph showing learnability, difficulty, and interest for user 2](image)

**Learnability:**

User rated by 2 for the tasks 2,3 for task 4 rated with 3 and for the task 5 users rated with 5. For tasks 2,3 user learned a little based on the tasks, but there was a gradual improvement for the task 4 and task 5. After all tasks, user felt that he/she was learning.

**Difficulty:**

User rated by 1 for the task 2,3,4 and for the task 5 user rated with 4. For task 2,3,4 user felt that the tasks were too easy but for task 5 users felt more difficult to perform. There was a huge difference compared to previous tasks.

**Interest:**

User showed the same level of interest for tasks 2,3 and then showed more interest for tasks 4,5. User rated level of interest for tasks 2,3 at 2 and for task 4 rated by 3 and continued same interest for task 5.
User3:

Figure 12 shows the results of learnability, difficulty and interest given user 1 for the tasks 2,3,4,5.

![User 3](image)

**Figure 12: Usability measures for user 3**

**Learnability:**

User rated with 4 for the task 2, for task 3 rated by 2, for task 4 with 1 and for the task 5 user rated with 2. From task 2 to task 3 and task 4 user’s learnability level decreased gradually and then for task 5 there was a little improvement compared to task 4. Overall user’s learnability started from a high level and in the middle dropped and then later increased a bit.

**Difficulty:**

User rated by 2 for the task 2, 3 for the task 4 rated with 1 and for the task 5 user rated with 4. For task 2,3 user felt the tasks were a bit easy, and for task 4 user felt too easy but for task 5 users felt more difficult to perform. There was a huge difference compared to previous task 4.

**Interest:**

User showed the same level of interest for tasks 2,3,4 and then showed a low interest for tasks 5. User rated level of interest for tasks 2,3,4 at 3 and for task 5 rated by 1. There is a huge difference from tasks 2,3,4 to task 5.
User 4:

Figure 13 shows the results of learnability, difficulty and interest given user 1 for the tasks 2,3,4,5.

![Usability measures for user 4](image)

**Learnability:**

User rated by 3 for the tasks 2,3,4,5. For tasks 2,3,4,5 users learned fairly well through all the tasks. The level of learnability is quite good for all the tasks.

**Difficulty:**

User rated by 3 for the task 2,3 for the task 4 rated with 1 and for the task 5 users rated with 5. For task 2,3 user felt the tasks were moderately difficult, and for task 4 users felt too easy but for task 5 users felt more difficult to perform. There was a huge difference compared to previous task 4.

**Interest:**

User showed the same level of interest for tasks 3,4,5, which was low after showing a bit interest in task 2. User rated level of interest for task 2 as 2 and for tasks 3,4,5 rated by 1. There is a no much difference in interest from task 2 to task 3,4,5.
User5:

Figure 14 shows the results of learnability, difficulty and interest given user 1 for the tasks 2,3,4,5.

![Figure 14: Usability measures for user 5](image)

**Learnability:**

User rated level of interest for tasks 3,4 at 3 and for task 2, 5 rated with 1. User rated with 1 for task 2, showed a same learnability level for tasks 3,4 and then felt he/she learns less for task 5. There was a gradual increase from task 2 to task 3 and user maintained same level of interest for task 4 and then lost interest in task 5.

**Difficulty:**

User rated by 3 for the task 2, rated with 4 for tasks 3,4 and for the task 5 user rated with 5. For task 2 users felt the task was moderately difficult, for tasks 3,4 user felt more difficult to perform, and then in task 5 users felt even more difficulty.

**Interest:**

User rated with 1 for task2, showed the same level of interest for tasks 3,4 and then showed a low interest for tasks 5. User rated level of interest for tasks 3,4 with 3 and for task 5 rated by 1. There was a gradual increase from task 2 to task 3 and user maintained same level of interest for task 4 and then lost interest in task 5.
4.1.7 User’s Overall Experience

Overall user experience is collected from the participants by providing the usability form at the end of the evaluation. The BLAST usability Form can be found in Appendix E. Based on the readings provided by the participants results were tabulated as in table 4. The fourth reading regarding the satisfaction is calculated from SUS. The overall average is considered. Radar chart is developed depending on the four readings.

<table>
<thead>
<tr>
<th>Difficulty (ease of use)</th>
<th>Interest (usefulness)</th>
<th>Ease of learning</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>58.4</td>
<td>43.2</td>
<td>46.8</td>
<td>46.8</td>
</tr>
</tbody>
</table>

Table 4: Readings for overall tool’s usability

Figure 15: Visualizing data using RADAR chart
From the chart we can interpret that overall tool’s usability in 4 ways.

- **Usefulness**: It helps me be more effective was productive and it is useful.
- **Difficulty**: It is hard and not user friendly, I cannot use it without written instructions.
- **Ease of learning**: I learned to use it quickly; I easily remember how to use it.
- **Satisfaction**: I am satisfied with it. It is pleasant to use it.

According to the figure 15, we can say that the overall BLAST’s usefulness is 43.2 %. Participants showed less interest in learning the things. The percentage of BLAST’s “Ease of use” is 58.4 %. Participants felt that the tasks performed were difficult. Both satisfaction and ease of learning hold same value i.e. 46.8 %. Participants were satisfied and felt they had learnt a little.

### 4.2 Qualitative Results

Results were collected through demographic questions; open-ended questionnaire, interviews and Expected measure. The questionnaire responses were described and analyzed. The observations were mainly focused on the participant’s interaction with BLAST’s user interface.

#### 4.2.1 Open-ended Questionnaire

Most questionnaires in usability studies include some open-ended questions, which allow the participants to add comments. In an open-ended questionnaire, participants were asked to answer the questions shown in Appendix F.

Positive comments that were given by the participants about the tool were:

1) Easy to use, the UI is good
2) Using several colors to show different errors
3) Easy to understand most parts of the interface and to edit test files

Negative comments that were given by the participants about the tool were:

1) Searching is confusing
2) Difficult to distinguish between colors
3) Some interface features that are hard to understand
4) Starting up the software
5) If you mark one word with several errors, you only see the latest error you added
Suggestions to improve that were suggested by participants were:

1) Search Function
2) Less options visible (Make more options available)
3) Perhaps adding some functionality to the search mode
4) If you select a word you should be given a list of all the error it has been marked with.

Issues faced by the participants during the evaluation were:

1) You cannot mark the space between two words for a missing word. Instead you have to mark the word before/after.
2) Sometimes the word that is marked is changed when you move into “edit mode”
3) I searched and counted errors manually. There should be a “global” search count.

Problems faced by the participants during the evaluation were:

1) I didn’t know the language rules/terminology very well, which made it hard to make correct annotations.
2) Advanced user interface with many options

Above mentioned comments and suggestion were from all the participants who participated in the evaluation.

4.2.2 Post-interview Results

This section provides the information regarding the answers given by the participants for the interview questions (see Appendix G) individually to the evaluator. The questions were asked orally and the answers were noted down in a written format. The results collected were shown as follows

Positive comments that were given by the participants when asked about the tool

1) Well developed, Good to use, Good UI
2) Ease of use, using several colors is good
3) Easy to understand and edit files
4) Easy to use and understand
5) All users felt that the UI is pleasant to use
Negative comments that were given by the participants when asked about the tool

1) Appropriate for linguists.
2) Starting up the software through a command prompt
3) Searching for errors is confusing
4) Not having a display of errors

Suggestions that were given by the participants when asked about what would they like to change in the BLAST?

1) Improved outlook
2) Improved search functionality
3) Adding help button
4) Distinguishing colors

Suggestions that were given by the participants when asked about how would they design the BLAST?

1) With some bright colors
2) Less menu options
3) Providing help and info buttons

User’s overall experience (Bad, good, very good)

1) Every user felt and answered the overall experience about the tool as “Good”.
5 Analysis

The Analysis section analyzes and discusses the issues observed during evaluation and overall experience of the tool. This section of report mainly describes the observations and issues observed by the users in two sections 5.1 and 5.2.

5.1 Observations

The observations in this thesis were mainly of specific nature, which focuses on analyzing the obtained data and on specific issues of how the participants interacted with BLAST. This section describes the results and user experience of usability study. Based upon the statements and feedback from the participants, as described in section 4, and my own opinions as an evaluator, I can summarize my own observations on the user experience as follows.

Observations based on tasks performed using BLAST

1) Ease of use in performing Task 1 & 3
2) Drastic change of annotation rate from task 3 to task 4
3) Lacking clarity in performing Task 5.

Explanation:

Participants were requested to perform 5 tasks. A summary of the user tasks was presented in the table 2 from section 3.3; see also in Appendix C. Out of these tasks, user’s performance was different from one another. Out of observations made, participants were extremely comfortable in performing task 1, as it was a simple task to upload a specific file through a command prompt. Task 3 is where participants were requested to annotate on a partially annotated file. After performing annotations in task 2, users got experienced a bit and they carried out the task 3 with better results. This was justified by the results they provided with expected and experienced measures for each task. (See table 3 from section 4.1). Average expectation measure for task 1 was 2 and participants experienced 2.2 and for task 3 the expectation measure was 2.2 and experienced measure was 2.

The lowest expectation and the highest experienced measure were found for task 5, which was intended to search for errors in a given file. Based on the results of this task, 3 users answered correctly out of 5 participants. However after performing tasks 1,2,3,4 participants got enough experience to perform task 5, searching errors of specific type.
and write the count. The expectation measure for this task was 1.8, which is low relative to other tasks, and unexpectedly its experienced measure was 3.8, which were the highest of all. A reason for this could be the participants’ level of confidence and was resulted in the highest measure.

Even though all the 5 tasks were performed and evaluated, out of them task 2, 3, 4 were treated to be the most important tasks of the evaluation. In section 4.1.2, from figure 6 it is evident that participant performed task 3 really well from the results obtained. When the values of annotations made by participants were compared to initial annotation's value there was a gradual increase in both levels of annotations i.e. 1-level and 2-level. Values were as follows initial value 0.5 and participant’s value 0.7 for 1-level and for 2-level annotations initial value 0.182 and participant’s value 0.6, which was considered as a big improvement. Task 3 was about doing annotations on a partially annotated file, which helped participants to perform correctly and confidently. But after comparing the values of task 4 to task 3 there was a drastic downfall in the values.

In section 4.1.2, from figure 7 shows that participants were failing to perform the task technically even though they finished the task on time. Task 4 was about finding and correcting the errors in a previously annotated file with error annotations. When the values of annotations made by the participants were compared to initial annotation's value there was a gradual decrease in both levels of annotations i.e. 1-level and 2-level. Values were as follows initial value 0.6 and participant’s value 0.4 for 1-level and for 2-level annotations initial value 0.6 and participant’s value 0.2, which can be considered as a big downfall. Finding the errors was about checking whether the words are properly tagged with an appropriate error type or not. So there was a chance of getting confused in between and end up with a wrong error type.

**Observations based on the user interface of BLAST**

1) Adding ease of use to search function
2) Using bright colors to distinguish between error types
3) Rearranging the error order (classification) with user friendly names
4) The UI looks pale and non-attractive
5) Deployment of help and info buttons
Explanation:

Based on the informal interviews and the questionnaire, adding ease of use to search function would be more help to the users while they search for some particular word or an error. The search function in the present version of BLAST is quite unclear while jumping from one annotation to another annotation to find errors of specific types. This vagueness is due to the user cannot recognize the error of that particular type in the sentence while sentences were changing.

Taking colors into consideration to support annotations, BLAST itself highlights different types of errors with different colors. But after evaluating the system, by interviewing about the colors associated with the errors and its impact on the participants, it is found that participants are not really aware of errors associated with colors and they did not even notice the color as the brightness of the color display looks different on Windows 7 operating system. Reason for this could be the predefined system color in Windows operating system is different from the Linux operation system. BLAST system was developed in a Linux environment and in this study evaluated on a Windows system. So this could be the technical reason for the participant not to notice colors supporting annotations.

There are 6 buttons to the bottom of BLAST that are not noticed and even not used by any participant. There was no task in evaluation to try and explore what those buttons do in reality. As per the feedback of participants, they feel that there is no need of those buttons, which do not have any obvious purpose. Even the names of those buttons Exact, LC_Exact, Word Form, Synonym, Paraphrase, Reset seem tricky to users.

Classification of errors was split into 5 categories as shown in the figure 2. The categories include “Missing words” “Word order” “Incorrect words” “Unknown words” “Punctuation”. These 5 categories are having some other sub categories. Based on the suggestions given by the participants, they feel that it was hard for them to annotate errors by correct classification. After finding the error they were unable to figure out to which category the error should fall in. The problem here seems to be that users lack knowledge of error classification or they are not aware of what actually category names meant.

When asked about the user experience of BLAST, there was a mixed response to the design of the user interface of the tool. Some of them said GUI is just satisfactory and some of them said GUI could be improved. When asked in specific to design and improvements, they said that UI design could be improved in a much more attractive way. They felt that UI is simple to use and easy to understand how the tool responds
but they did not like the way the tool look like. They suggested having attractive colors for foreground and background makes the user feel fresh and interested while using.

Users felt that the tool functionality is a little incomplete without Help and Info buttons. There are no buttons or such functionality where the user can get a help or required information to understand what they are doing. Participants wanted to have a help function, which tells in which mode they actually are operating in and info about the tool and its developer details and releases.

5.2 Issues and Recommendations

After analyzing the evaluation results and the feedback got from the users, several issues were found. This section presents the usability issues found, their severity and my recommendations to overcome them. Tullis & Albert (2008) was followed to rank the severity of issues.

Issue 1: Trouble finding the errors in Search mode

Severity: Medium.

Description: Finding the errors of a specific type of a previously annotated file and telling the count of the errors in a search mode was one of the tasks. Users found it difficult to understand how the search mode work, how to identify the type of errors and where to count and how?

Recommendation: By providing some information regarding how to operate the tool in a Search mode through a popup screen or a help button when clicked could be helpful. Instead of navigating from sentence to sentence and search for errors and counting them, it would be better to provide a search button with some functionality to search for one particular type of error in an annotated file and display the count of errors in a separate text box. By following above-mentioned recommendations the severity of this issue can be reduced.

Issue 2: Same options available in all modes of operation

Severity: Low.

Description: There are several options available in BLAST to operate. The problem is all options are not active and visible in all modes of operation. Users felt that there is no use of having access to all options when operating in different modes. When the user was about to edit or remove annotation, by clicking one other option by mistake leads to start the process again from the start.
Recommendation: By providing separate options according to the menu could be helpful. Appropriate options and functionality for the tool depending upon the mode of operation and complementing the user’s needs could be helpful. By having a separate way of highlighting available options or by separating the options according to mode will be helpful. By following above-mentioned recommendations the severity of this issue can be reduced.

Issue 3: Annotating on the previously annotated file or changing annotations.

Severity: Medium.

Description: When a user was asked to annotate on a previously annotated file or edit the existing annotations, user was ambiguous what he/she was doing. The user wasn’t sure to annotate by removing, re-annotate and update.

Recommendation: By providing some information regarding how to edit the annotated file in edit mode through a popup screen or a help button when clicked could be helpful. By providing this information, the user clearly understands what to do and how to operate in edit mode. It would be better to provide an update button with some functionality to update for overwriting the error type in an annotated file and highlight the error with an appropriate color in the sentence. By following above-mentioned recommendations the severity of this issue can be reduced.

Issue 4: Low learning experience

Severity: High.

Description: When a user was asked to perform some tasks in the tool and rate his/her level of learnability, the user’s learnability level was less. There was no satisfaction for the user that actually he/she learnt something.

Recommendation: By providing some information regarding how to upload and annotate the uploaded file in annotation, edit and search mode through a popup screen or a help button when clicked could be helpful. By providing this information, the user clearly understands what to do and how to operate in annotation, edit and search mode. It would be better to provide some information how to startup and access the tool for beginners will be helpful. By following above-mentioned recommendations the severity of this issue can be reduced.
Issue 5: Task 5 in “Fix it fast” quadrant.

Severity: Medium.

Description: When a user was asked to perform a search for errors task on the tool, his/her level of expectation was too low. But the user’s experience on that particular task was too high which kept that task to be in “fix it fast” quadrant.

Recommendation: By explaining the task’s description properly could reduce the expectation of the task to be less. By providing some information regarding how to operate the tool in a Search mode through a help button when clicked could be helpful in improving the user experience. By following above-mentioned recommendations the severity of this issue can be reduced.

Issue 6: Use of extra buttons (Exact, LC_Exact, Word form, Synonym, Reset)?

Severity: Low.

Description: When a user was asked to explore the tool and all of its options available, he/she experienced confusion. The user’s experience and feedback on this particular issue was, why and when to use these extra buttons? Participants thought that these buttons were of no use and wanted to remove them.

Recommendation: By explaining the button’s usefulness by a proper description through help or info button could reduce the confusion of the participant. By providing some information regarding how to operate the tool with these buttons through help button when clicked could be helpful in improving the user experience. Or by taking out the buttons they were of no use without disturbing the main functionality of the tool will also be useful. Hiding the buttons provided based upon the usage of the tool and task could also be helpful as these buttons could be needed and useful for other tasks if we use the tool full-fledged. By following above-mentioned recommendations the severity of this issue can be reduced.
6 Redesign Proposals

This section mainly discusses the design proposals. The main idea is how the interface of BLAST can be redesigned to overcome/minimize above issues mentioned. The recommendations and suggestions from participants were considered in the proposal of redesign. As mentioned above in chapter 3.7 design processes was followed in 2 phases i.e. initially drawing paper-based prototypes and later implementing in tool-based prototypes. The screen shots of prototypes are provided below. Before going in to the details of the tool-based prototypes here are the original screenshots of BLAST in all three modes 1) Annotation mode 2) Edit mode 3) Search mode as shown in figures 16, 17, 18.

Figure 16: Graphical User Interface of BLAST (Annotation mode)
Figure 17: Graphical User Interface of BLAST (Edit mode)

<table>
<thead>
<tr>
<th>Annotation mode</th>
<th>Edit mode</th>
<th>Search mode</th>
<th>Annotation count:</th>
<th>Current annotation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Previous</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Next</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remove</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exact</td>
<td>LC_Exact</td>
<td>WordForm</td>
<td>Synonym</td>
<td>Paraphrase</td>
</tr>
</tbody>
</table>

Figure 18: Graphical User Interface of BLAST (Search mode)
6.1.1 Proposal of Designs

Proposal 1:

Proposal 1 follows the same structure as the old version of BLAST but with slight modifications like adding help button on the top, buttons to navigate between sentences and fading out the mode which the user is actually working in. Compared to the original version of BLAST the buttons at the bottom as shown in figure 18 were removed for the convenience of the users. Buttons were removed to overcome the issue 6 mentioned in section 5.2. In annotation mode, out of all the three options annotate,
change, remove buttons only annotate option can be accessed, but in edit mode as shown in figure 20 only two options are available.

![Figure 20: Proposal 1 (edit mode)](image)

In edit mode, users can view two options change and remove buttons. In this mode he/she can change/remove the annotations made. User can switch easily from option to another. Navigation button, which was called as “Error navigation”, was also provided in this proposal to go forward and backward among the annotations made for errors in the sentences. There is also the option of navigating by clicking next and previous buttons, which were just displayed below the sentences to navigate between the sentences. There were buttons to change/remove as shown in figure 17, but in this proposal there are separate buttons to change/remove, which makes the user to edit the annotated files easily and properly. Help button is also provided to give basic information how the tool operates in edit mode for the user when clicked.
In search mode, users can search for errors by selecting on the error type from the classification and pressing the search button. The error count of the errors annotated will be displayed in the text box available on the screen as shown in figure 21. Sentences containing errors can be navigated by using the navigation buttons below the search button. User can get help by clicking on the help button. A text explaining how and what to do will allow users to perform this task easily. Options like Error count and navigation between annotations were not provided in the older version of BLAST as shown in figure 18. In original version of BLAST it is hard to search for the errors until unless the user is aware of how the errors are represented with different colors.
Proposal 2:

Figure 22: Redesign proposal 2

There is no big difference between proposal 1 and proposal 2 except the additional 6 buttons, which were removed for user’s convenience in proposal 1, was replaced here in proposal 2. Help button and the fading out ideology (when clicked on a button, the background looks like fading out) were common. The figure of edit mode and the search mode were not shown here in proposal 2. To have the basic idea of modes of operation refer to figure 20 and figure 21.
Proposal 3:

In proposal 3, all the modes of operation were placed in a cascading style horizontally. Annotate, edit and search buttons were placed side by side as shown in figure 23. Help button was placed in the top left of the tool window. A navigation button was placed instead of traditional previous and next buttons at the bottom of the window. This is used to navigate between the sentences in a file. Button heads of SRC, SYS and REF were highlighted with a black outline. The mode of operation was highlighted with orange color, which makes the user to get aware of which mode is operating in.
Figure 24: Redesign proposal 3(edit mode)

From the figure 24, we can see that in edit mode, users can view two options update, update and remove buttons. In this mode he/she can annotate change (update) /remove the annotations made. User can switch easily from option to another. Navigation button was also provided in this proposal to go forward and backward among the sentences. There is no option of navigating by clicking next and previous buttons, which were just displayed below the sentences like in proposal 2. Here in this edit mode update button can be used as update to overcome the issue 3 (annotating on the previously annotated file or changing annotations) discussed in section 5.2.
In search mode, users can search for errors by selecting on the error type and pressing the search button. The error count will be displayed in the text box available on the screen. Sentences containing errors can be navigated by using the navigation buttons below the search button. Next and previous button are used to navigate between the errors in the annotated files. These functionalities were included because these are some of the issues found by the participants during evaluation. See for issue 1 (trouble finding errors in search mode) in section 5.2. Help was also introduced to help the user by popping up a textbox when clicked on help button as shown in figure 25.
Actually in previous version of BLAST, it is possible to enter the error type, but none of the participants came to know about its existence. So now a separate label and a text box to enter the error type was introduced. In search mode as shown in figure 26, users can search for errors by selecting on the error type or entering the name of the error type in the text box given to label Error type and pressing the search button. The error count will be displayed in the text box available on the screen. Sentences containing errors can be navigated by using the navigation buttons below the search button.
6.1.2 Modifications

After discussing why, how and what about the BLAST’s interface and its mode of operation, it is equally important to state the modifications suggested by the evaluator after the evaluation. There were some ideas adapted and certain choices made by the evaluator in this evaluation and explained in detail.

Modifications:

The modifications suggested by evaluator in this interface re-design were

Modification 1: Introducing “Help” button.

Description: As an evaluator, I wanted to introduce the “Help” button and integrate it to the interface of BLAST. Help button generates a pop-up window with some text, which contains information about how the tool was operated in different modes. This is due to the observations made by me during the evaluation and feedback given by the participants. In this evaluation, participants experienced problem to understand the intended use of buttons. They were also unclear what and how to perform for the tasks given in different modes. So to overcome above addressed problems help was used. Now by introducing the help button user can know in which mode they are working and how to perform task in that particular mode of operation. User on clicking the help button a pop-up window will be generated and gives the description about the mode of operation. For example see figure 26 a popup window having text explaining what to do in Search mode. This choice was made to make the participants properly perform the tasks without any ambiguity.

Modification 2: Horizontal display of modes of operation.

Description: After going through the results obtained from participants as mentioned in sections 4.2.1 and 4.2.2, I wanted to show the operation in a horizontal order instead of vertical order on the interface of BLAST. In this evaluation, participants experienced problem to use of buttons in three modes of operation as all the buttons looks alike. They couldn’t distinguish of buttons for the switch of modes and buttons for performing tasks. They were also risking to hit the wrong button while performing he task. Above addressed problems can be overcome by introducing the buttons in horizontal order and highlighting the mode of operation with orange color background. User can know in which mode they are working and there is a less chance of doing mistakes. For example see figure 23 all mode buttons were placed on the left and buttons to perform tasks were placed on the right. This modification makes the participants not to click different buttons while performing tasks.
Modification 3: Displaying “Error” count.

Description: As an evaluator, I wanted to show the “Error count” text box and label to display the count of errors. This modification was introduced due to the observation made by me during the evaluation and feedback given by the participants in section 4.2.2. In this evaluation, participants weren’t able to count the specific type of errors from the annotations made on a specific file. User has to manually count for the particular types of errors based on the colors associated to the errors. So to overcome above addressed problems an error count was used. Now by introducing the search button associated with error count functionality, user can know the count of errors just by selecting the error type. When clicking the search button the count of errors will be displayed in the textbox associated. See figure 21, which shows error count label and count of errors and a search button. This choice was made to make the participants properly search for the error types and count for the errors of specific type.

Modification 4: Different types of display for sentence navigation and error navigation.

Description: Two types of displays were introduced to represent the navigation between the sentences and errors from the annotated files. There were these features in original version of BLAST, which wasn’t recognized by the participant’s during evaluation. There were also no separate button styles to recognize which buttons were used for what purpose but now in the redesign proposal 2 and proposal 3, sentence navigation has standard next and previous buttons to navigate between sentences and a button with arrows to navigate between errors in an annotated file. By this slight modification users can make a choice of using buttons depending on the task.

6.1.3 Follow up interview on redesigns

An informal follow up interview was conducted after several redesigns were proposed. Proposed redesigns are being shown to two of the participants and asked for their feedback. The participants felt quite happy with the redesigns as they can see the improvements they had suggested in informal interviews. Three participants were not interviewed, as they cannot make their presence there at the interview.
7 Conclusion

In this thesis I presented an evaluation of BLAST, an error analysis tool. I conducted the usability study to find the design and usability issues of the tool. Three users in the pilot study and five users in evaluation tested/used the tool in a time of 45 minutes. The main focus of the evaluation was on user acceptance and the usability. Several methods were considered and used during evaluation to collect information.

After the whole evaluation process, the goal of the evaluation study was fulfilled as this study evaluates the user experience (learnability, difficulty, interest and satisfaction) of BLAST. User satisfaction and usability issues of BLAST were found.

There was some usability issues found, one of them was to operate the tool in the search mode, which was explained in the evaluation study, and recommendation was also made to overcome. The recommendations were implemented as a design draft or prototypes, which represents the UI of BLAST. The other issue found, was the degradation of User’s performance in annotating from one task to another. There were problems with the annotations made by the users when compared to Gold Standard evaluations. The users failed to perform the task 4 in terms of success rate compared to task3.

The suggestions given by the participants were analyzed and later implemented in developing prototypes (redesigns). Redesign proposals were demonstrated to two participants from the evaluations study and were asked for feedback and suggestions. Feedback of participants at this level was vital and they felt that the interface looks comparatively good and better options are available in design proposals.

To know how good the redesigns proposals when compared to the original design, I managed to get a feedback by looking at the interface and the comments given by participants. But to know the difference in exact and actual, there is a chance of doing further work to evaluate the redesigns and then compare the results to my findings. Adapting new designs to the existing working software or building up a new system by taking participants feedback in to consideration can do further study. A usability evaluation study should be followed on the newly developed system as well.
8 References

Appendices

Appendix A: Test Introduction paper

The test that you are about to take is a part of graduate thesis in Computer Science at Linköping University. The test is designed to evaluate the usefulness of software called BLAST.

BLAST: An open source tool for error analysis of machine translation output developed by Sara Stymne. It is a graphical user interface, which can be used with any Machine Translation System. BLAST can aid the user by highlighting similarities with a reference sentence.

Participation in the test is completely voluntary and you can cancel at any time. All information collected will remain anonymous. The consolidated results of the study will be published at Linköping University.

The test is conducted in person. It consists of several tasks that you will be performing. Each task should be done in a certain time limit. When you do a task, continue until you either finished or reached the point where you are stuck. The evaluator will not answer any questions during the test, besides clarification of the instructions.

If you are unsure how to proceed with a task, please read the instructions again!

Instructions:
• You will be provided with 3 files and a TESTID.
• Provided files are copied to a working directory C:\Blast-v1.
• You should use a command prompt to open the required file.
  o Command for opening a file:
    java -jar blast-1.0.jar -c lib/sw-detailed-cats -a Test1.
• You have to perform tasks given in the pages 3, 4, and 5.
• You have to save the results by giving the name as testid_taskno.
• User was requested to provide written feedback.

Thank you for participating in the study!
Appendix B: Demographic Questions

Gender:

F: Female      M: Male

Age: To which age group you belong to?


Education:

What is the highest level of education you have completed?

High school graduate
Technical (vocational) certificated
Bachelor degree
Master degree

Experience:

How good are you at reading and understanding English?

Good      Very good      Excellent

How good are you at writing English?

Good      Very good      Excellent

How often do you read English (Newspaper, Articles, Novels, and Books)?

Daily      Weekly      Monthly

How many years of English study do you have?

1 – 5      5 - 10      more than 10 years

Do you have previous knowledge of evaluating an interface?

Yes      No

If yes, how many times have you evaluated?

0      1      2      3      4      5      more than 5 times

Have you attended any of Usability related courses before?

Yes      No
Appendix C: Experiment Tasks

For each of the following statements, check a box that best describes your reactions based on the task.

Task 1: Upload and run a given blast file named “test 1” through command prompt.

Before: How easy do you expect the task to be?

<table>
<thead>
<tr>
<th>Very easy</th>
<th>Very hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5</td>
<td></td>
</tr>
</tbody>
</table>

After: How easy was the task to perform?

<table>
<thead>
<tr>
<th>Very easy</th>
<th>Very hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5</td>
<td></td>
</tr>
</tbody>
</table>

What are the issues found?

Task 2: Make annotations on the uploaded “test 1” file.

Before: How easy do you expect the task to be?

<table>
<thead>
<tr>
<th>Very easy</th>
<th>Very hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5</td>
<td></td>
</tr>
</tbody>
</table>

After: How easy was the task to perform?

<table>
<thead>
<tr>
<th>Very easy</th>
<th>Very hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5</td>
<td></td>
</tr>
</tbody>
</table>

What are the issues found?

Task 3: Make additional annotations on a given partially annotated file (test 2).
Before: How easy do you expect the task to be?

<table>
<thead>
<tr>
<th>Very easy</th>
<th>Very hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

After: How easy was the task to perform?

What are the issues found?

Task 4: Find, remove and change errors in the annotations made in a previously annotated file (test 3).
Task 5: Find out how many word order errors exist in an annotated file (test 3)?

Before: How easy do you expect the task to be?  
Very easy       Very hard
1  2  3  4  5

After: How easy was the task to perform?  
1  2  3  4  5

Number of errors found?

What are the issues found?
**Appendix D: The System Usability Scale**

**Instructions:** For each of the following statements, check a box that best describes your reactions to the BLAST.

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I think I would like to use this system more often.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>I thought the system was unnecessarily complicated.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>I thought the system was easy to use.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>I think I would need help of a technically qualified person to use this system.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>I think that the different functions in this Systems are well coordinated.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>I thought there was too much Inconsistency in the system.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>I imagine that most people would. learn to use this system very quickly</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>I thought the system was very cumbersome to use.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>I felt very confident when I used the System.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>I had to learn things before I could get started with the use of the system.</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
Appendix E: The BLAST Usability Form

Complete the usability form individually at the end of the evaluation session. Your feedback is important in improving the performance of BLAST with respect to efficiency, effectiveness and satisfaction.

Instructions:

For each section, please rate the following (range 1 to 5 in all cases)

- Difficulty: Rate the degree of difficulty (1 = too easy, 5 = too difficult)
- Learning: Rate your learning experience (1 = learned nothing, 5 = learned a lot)
- Interest: Rate your interest level after completing the task (1 = no interest, 5 = high interest)

<table>
<thead>
<tr>
<th>Task</th>
<th>Difficulty</th>
<th>Learning</th>
<th>Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make Annotations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change Annotations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remove Annotations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search for Error Annotations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix F: Open-Ended questionnaire

Please answer the following questions:

• What did you like about this tool?

• What did you dislike about this tool?

• What would you like to suggest improving?

• Was there anything during the evaluation you thought was problematic? For example, something that was complicated, redundant, missing, or if you have any other comments.
Appendix G: Interview questionnaire

1. What were your impressions of the BLAST?
2. Was there anything you felt bad?
3. Was there anything you felt good?
4. How was your experience? Opt for one of three (bad, good, very good)
5. What would you like to change in the BLAST?
6. Was the UI pleasant to use? Do you like it?
7. What changes do you want to have on BLAST?
8. How would you design the system