Semi-automatic Multimodal Web Content Retargeting System

by

Cansu Sen

A Thesis Submitted to the
Graduate School of Sicences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science
in
Computer Sciences and Engineering

Koc University
March 2016

Koc University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Cansu Sen

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Assist. Prof. T. Metin Sezgin (Koç University)(Advisor)

_____

Assoc. Prof. Engin Erzin (Koç University)

_____

Assoc. Prof. M. Elif Karslıgil (Yıldız Technical University)

Date: _____

ABSTRACT

Non-desktop platforms such as mobile phones, tablets, and connected TVs now serve as primary means for browsing the web. Unfortunately, much of the existing web content has been designed and optimized for desktop viewing. Hence, tailoring the content for viewing on multiple platforms with various constraints has become a necessity. However, manual customization for each platform is costly, and not always feasible. In this work, we introduce a semi-automatic multimodal web content retargeting system that works with minimal assistance of an expert. From a single page, our system learns how a web portal should be tailored based on verbal and gestural descriptions of a human operator, and converts previously unseen pages based on what it has learnt. We have three contributions. First, we introduce a modular system architecture with a flexible backend for learning content translation rules from examples using machine learning. Our second contribution is a translation engine capable of applying previously learned translation rules to translate unseen pages. Our third contribution is a large database consisting of 300 manually coded web pages from 10 popular portals, which will serve as a benchmark database for evaluating similar systems in the future. Our discussion is supported by objective evaluation of the system's ability to learn, and results from a carefully designed usability study demonstrating the effectiveness of the proposed system with real users. The results show that our system can learn conversion rules and it is a usable system.

ÖZET

Günümüzde akıllı telefonlar, tablet bilgisayarlar, hatta akıllı televizyonlar giderek daha güçlü işlemcilere ve hesaplama gücüne kavuşmaktadırlar. Bilgisayarlar tarafından gerçekleştirilen birçok görev, örneğin internet kullanımı bu cihazlarda da aktif olarak gerçekleştirilebilir duruma gelmiştir. Ancak mevcut internet sitelerinin büyük bir çoğunluğu masaüstü bilgisayarlarda görüntülenmek üzere tasarlanmıştır. Bu nedenle her cihazın donanımsal özelliklerine uygun web tasarımlarının oluşturulması bir zorunluluk haline gelmiştir. Ancak her internet sitesinin manuel olarak yeniden tasarlanması çok yüksek miktarda uzman analizi ve maliyet gerektirmektedir. Bu çalışmada, bahsi geçen problem için yeni bir çözüm olarak, yarı otomatik çok-kipli internet içeriği çeviri sistemi geliştirilmiştir. Geliştirdiğimiz sistem, tek bir web sayfasından, bu sayfanın farklı bir platform için nasıl yeniden tasarlanması gerektiğini, bir uzmanın sözlü tanımlamalarına dayanarak öğrenebilir ve öğrendiği bu modele göre sayfanın farklı örneklerinin yeniden dizaynını gerçekleştirebilir. Geliştirdiğimiz sistem literatüre üç temel katkı sağlamaktadır. İlk katkımız, makine öğrenmesi metotlarını kullanarak internet siteleri için çeviri modelleri öğrenebilen modüler bir sistem mimarisidir. İkinci katkımız, öğrendigi modelleri kullanarak daha önce görülmemiş sayfaları yeniden dizayn edebilen bir çeviri motorudur. Üçüncü katkımız ise, benzer sistemlerin değerlendirilmesinde kullanılabilecek, 10 farklı web portalının 300 örneği kullanılarak oluşturulmuş bir veri arşividir. Sistem üzerinde gerçekleştirdiğimiz nesnel değerlendirmelere göre, geliştirdiğimiz sistem çeviri modellerini yüksek bir başarı oranı ile öğrenebilen bir sistemdir. Nesnel değerlendirmelerin yani sıra, gerçek kullanıcılar ile yapmış olduğumuz kullanıcı deneyleri, sistemin kullanılabilir olduğunu göstermiştir.

ACKNOWLEDGEMENTS

I would like to thank my advisor Asst. Prof. Tevfik Metin Sezgin for his valuable guidance and support throughout my studies.

I am indebted to Assoc. Prof. Elif Karslıgil and Assoc. Prof. Engin Erzin for agreeing to be in my thesis committee and contributing with their suggestions and comments.

I am forever grateful to my understanding family; who endured this long process with me, always offering support and love.

I would like to give special thanks to Mert Demirer, for always standing by me. His advice and enthusiasm helped me do my research with greater motivation.

I owe many thanks to all members of IUI lab, Neşe Alyüz, Çağla Çığ, Kemal Yeşilbek, Berker Türker, Shabbir Marzban, Özem Kalay, Banuçiçek Gürcüoğlu, Burak Özen, Deniz Can Yıldırım and Ozan Can Altıok who listened to my research for their times and valuable feedback. I also would like to thank Alkin Kiroglu for all his helps.

Finally, most special thanks to Şerike Çakmak, who spent almost as much time as me for this study, helped me millions of times with her valuable comments and sincere efforts.

This thesis is funded by SAN-TEZ under grant number 01198.stz.2012-1.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

KNN	K Nearest Neighbor
SAPI	Speech Recognition API
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
TR	Translation Rule
SRGS	Speech Recognition Grammar Specification

Chapter 1

INTRODUCTION

Devices such as smart TVs, mobile phones and tablet PCs have evolved into powerful computers. They now possess larger screens, more processing power and more importantly, Internet connections. Much of the web browsing has moved away from traditional desktop computers to these platforms. However, the vast majority of web content is designed for desktop viewing and this causes layout and usability problems when they are rendered on non-desktop platforms. These problems have three main causes. First, screen sizes vary substantially across devices. Different screen sizes cause inferior rendering of web pages. Second, the distance of the user from the screen differs for each device. For instance, a web page designed for desktop PCs rendered on a TV will not be readable because of the distance of the user from the screen. Third, interaction technology causes limitations. Most websites are designed for mouse and keyboard interaction. Remote controls, for instance, restrict the usage of some features. All of these facts create a necessity for custom designs for each website and for each device.

One solution to this problem is to manually create and implement new designs from scratch for each website. However, this costs money and time. Another option is to use automatic conversion methods that have been suggested by different researchers over the last decade [2] [3] [5] [6]. Automatic translation, however, is difficult and results are still far from being satisfactory.

We offer an alternative solution based on a simple observation: The content of websites changes frequently, but the main structure of the pages remains the same. If a design expert were to examine an instance of a web page and explain a new format to another human, the human could easily understand this new format and apply the same changes to other instances of the same website in order to acquire converted versions of that page. Can computers do the same thing? Based on this motivation, we designed a system architecture which combines human and machine intelligence and outputs high-quality versions of existing web pages and implemented this design.

Our system requires minimal expert involvement to acquire human expertise on web design. It then combines human expertise and knowledge gained from the previous days of a web portal to build a translation model specified to a platform. An independent component of the system generates adapted version of any instance of the web portal based on the previously saved translation model. We also evaluated the system in terms of usability and ability to learn.

Our system has three novel approaches. First, it employs a semi-automatic conversion method to create more usable adapted pages. Second, it uses machine learning to learn translation models and parts of web pages. Third, it uses multimodal interaction to offer a natural communication with users.

The rest of this thesis is organized as follows. Section 2 investigates the related work. Section 3 describes the architecture and the components of the system. Section 4 explains the multimodal aspects of the system. Sections 5 and 6 describe the evaluations of learning and usability respectively. Section 7 concludes the thesis.

Chapter 2

LITERATURE REVIEW

Ever since hand-held devices were equipped with the Internet connection, researchers have been trying to find ways to view web pages in a better way on these devices without additional programming efforts. Hence, web content retargeting is a well-explored research area.

The vast majority of the forerunner research focuses on developing automatic systems which requires an understanding of the semantic structure of a web page. While the way of discovering the structure of the web page changes, the general idea remains the same in these works. They first identify the semantically consistent blocks of a web page, and then retarget those blocks to create the adapted web page in the runtime. The most likely drawback of these systems is retargeting inconsistent or unreasonable blocks. Also, they usually focus on small screen devices, hence, the primary consideration is to eliminate the horizontal scrolling and to leave only the vertical scrolling on the page. Hwang et al. [1] propose representing the web page as a tree and forming subtrees out of consistent blocks. They form new pages using each subtree and omit these parts from the original web page. Instead, they place hyperlinks which direct viewers to the new pages. Thus, they simplify the original web page. Similarly, Chen et al. [2] use HTML DOM tree to split the original page into subpages. They use a rule-based splitting algorithm based on HTML tags. Lee et al. [3] make use of the VIPS Algorithm [4] to identify the semantically related blocks. They then extract titles for each block and also summarize each block to show only the extracted

titles and summaries on the main page. There are more studies in the literature [5] [6] [7] that work in the similar ways with small differences.

One of the most crucial problems with focusing only desktop to mobile adaptation is that those methods are not easily extended to other contexts such as smart TVs [18]. Adaptation for large screen contexts differs from adaptation for small screen contexts in two major ways: 1) The majority of the websites fail to support large screen context, therefore, there is a need for adaptation. 2) Screen size, resolution and supported input and output modalities are quite different and this creates new challenges for adaptation for large screen context [9]. A recent work [9], whose focus is large screen devices, develops a context-aware web design tool. It then uses crowdsourcing to use this tool and generate many customized layouts.

Another recent work [10] suggests a hybrid approach to adapt web pages. They combine three popular adaptation techniques: tree-view, hierarchical text summarization, and colored keyword highlighting. However, they also use automatic translation.

There exist some works which include human intelligence into the adaptation process. [11] first analyze the structure of the web page and generate a series of HTML blocks. They then ask users to examine the results and rank the possible retargeting blocks. Their work is also limited to small screen devices.

Our system does not suffer from the existing limitations. The first reason of this is that our system does not propose a device-specific solution. It can create translation models specialized for different devices. Moreover, thanks to the flexible architecture of the translation rule library, many rules targeting different setups can be added to the system to make the output page more qualified for a specific context.

The second reason is that our system uses human input which guarantees to preserve the most important parts of the pages in the adapted versions. This is because a human makes design-related decisions, instead of a machine.

In addition to proposing a new solution to the web content retargeting problem, another focus of our work is to create a multimodal intelligent user interface which uses speech and sketch modalities for interaction. Multimodal interaction has been gaining popularity in recent years as it offers a more powerful way of communicating with computers. In literature, there are many works which build multimodal interfaces with speech and sketch modalities. Blessing et al. [14] build an interface for interactive road design and use speech and sketch as interaction modalities. Rossetto et al. [15] propose a video retrieval engine which enables users to build queries by sketching and talking to the system. Hofmann et al. [19] build a speech-based in-car human machine interface. Our work also contributes the literature as being a multimodal interaction effort through speech and sketch modalities.

Chapter 3

METHODOLOGY

Our system consists of two main parts and two auxiliary parts that provide necessary rules and functions for the main parts. The first main part is the multimodal intelligent user interface. It uses human expertise and previous days' web pages as inputs and creates translation models for web portals. The second main part is the translation engine. It uses the translation models created through the first part and an unseen web page instance to generate the converted version of it. Preprocessed web page instances are stored in the web archive, the first auxiliary part of the system. Translation rule library, the second auxiliary part, defines specialized rules each of which implements HTML code modifications to build web pages. Both main parts of the system have access to the web archive and the translation rule library. The architecture of the system is shown in Figure 3-1. This section explains the parts of the system and the processes that take place between these parts.

3.1. Web Archive

As our first step, we compiled a web archive consisting of a variety of instances of different websites. The system requires such an archive for two reasons: 1) The system uses machine learning to learn translation models. This process requires previous days' information of a website to achieve a higher accuracy. 2) The system also uses sketch interaction for selections on a web page. This process requires a prior knowledge of web page to resolve the ambiguity of selections.

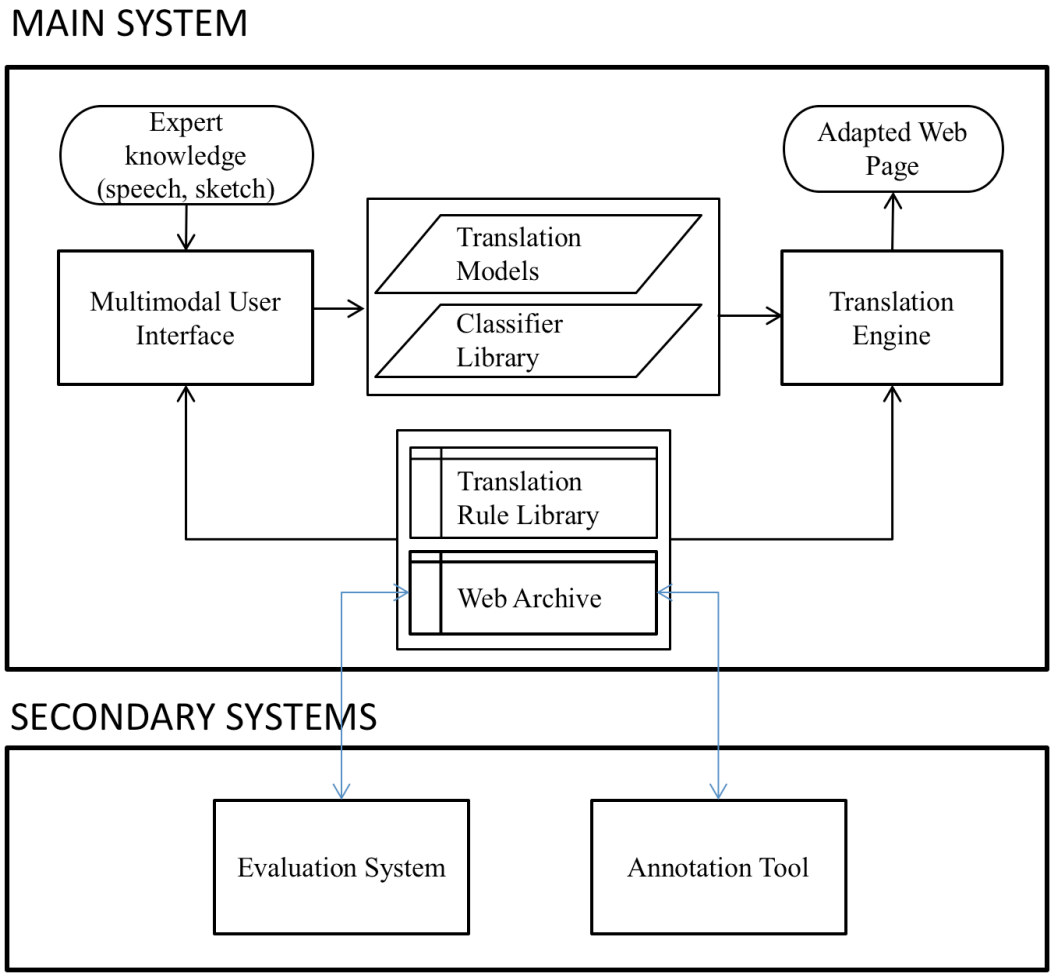


Figure 3-1: The main parts of the system and the interaction between them.

We formed the web archive in the following way. To resolve the ambiguity of a selection on a web page, the system necessitates identification of HTML elements of a web page. An HTML element is an individual component of an HTML document, which consists of a start tag, an end tag and content in between. We retrieved all the HTML elements of a set of web pages. Figure 3-3 shows the organization scheme of the web archive.

Our web archive contains 10 to 30 unique pages of 10 different websites. These websites are listed in Table 3-1. Pages were retrieved within a period of 8 months (February 2014 September 2014) on random days. We saved the screenshot, HTML, text and position information of each HTML element of each web page. Figure 3-2 shows a snippet of a web page. For the red framed HTML element in Figure 3-2:

- HTML information is the HTML code of the red squared area:

```
<div>
  <a href="…"> <img src="…"/></a>
  <a>Save elephants: Give jobs to their killers</a>
</div>
```

- Text information is the text that appears in the area:

  "Save elephants: Give jobs to their killers"

- Position information consists of the X and Y coordinates of the upper left corner of the rectangular area along with the width and height values.
- Screenshot information is the image file of the area saved in jpg format.

Figure 3-2: A snippet of a web page. Framed area shows an HTML element.

Table 3-1: The web archive contains pages of these portals.

Websites
www.about.com
www.ask.com
www.cnn.com
www.ebay.com
www.theguardian.com/fashion
www.imdb.com
www.recipe.com
www.reuters.com
www.scientificamerican.com
www.unicef.org

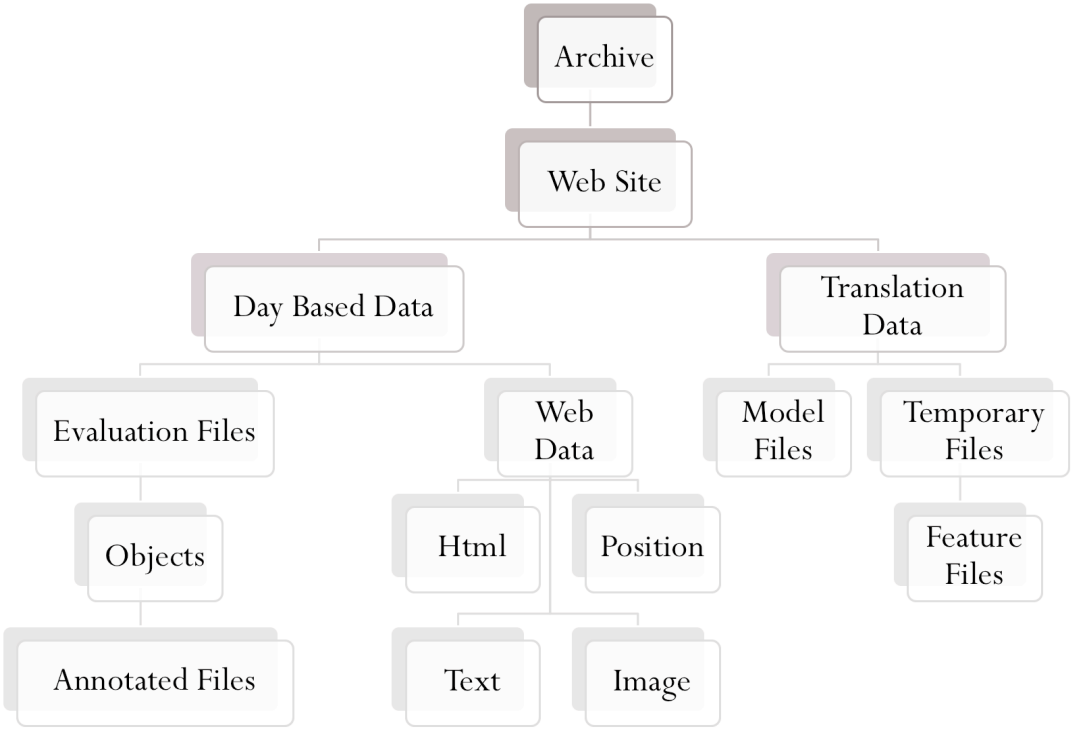


Figure 3-3: Hierarchical organization of the web archive

3.2. Multimodal User Interface

The multimodal user interface is the part through which the expert interacts with the system and describes the design of the adapted web page. The expert creates translation models based on a source HTML page which is retrieved from the web archive. The multimodal user interface contains two HTML rendering engines. The first engine allows the expert to view the source HTML pages and the second engine renders the adapted web page as it is formed. Figure 3-4 shows the design of the multimodal user interface. The expert interacts with the system through the speech and sketch modalities. Every action of the expert is resolved through a reference resolution system.

Figure 3-4: Design of the multimodal user interface.

3.3. Reference Resolution System

When speech is detected by the multimodal user interface, the reference resolution system interprets the speech and matches it to one of the translation rules in the translation rule library. Translation rules are explained in Section 3.5.

The expert is expected to teach the system which parts of the source page will be retargeted for the new design. The expert does this by sketching around these parts on the page.

When a sketch is detected, the system identifies the corresponding HMTL element indicated by the sketched area by using the position and size information previously calculated for each element of each page in the archive. To compare the sketched area and possible elements, an algorithm first identifies the topmost, bottommost, leftmost and rightmost sketch points. Then the area takes place between these points and the area of HTML elements are compared. The algorithm first finds the best match and then searches for other elements that lie within a threshold distance to the best match. We observed that if there is more than one element which fits this criterion, it is most likely that the elements are the same, i.e. having the same screenshot with an additional non-functional HTML tag (e.g. a div tag) in the HTML code. In such cases, we selected the element which has the longest HTML code to guarantee that it contains the area indicated by the expert.

The selection of the expert can either contain a single HTML element, or it can be the union of numerous HTML elements. Example sketches and corresponding HTML elements can be seen in Figure 3-5.

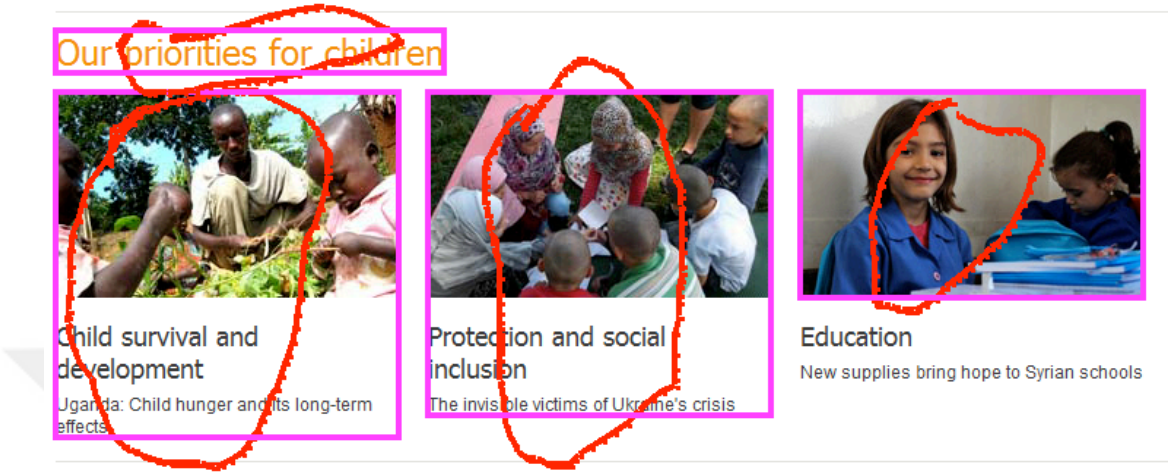


Figure 3-5: A web page snippet from unicef.org and the corresponding HTML elements for sketched areas. Red drawings show sketches done by the expert, pink drawings show the corresponding HMTL elements calculated by the system.

3.4. Web Element Learning

In order to place an HTML element in the adapted page, the element has to be learned and correctly predicted. For this reason, we established a web element learning process.

Web element learning is used by the two main parts of the system, the multimodal user interface, and the translation engine. The multimodal user interface trains a classifier for each HTML element which is going to be retargeted and saves the resulting classifiers. The translation engine makes predictions based on previously saved classifiers.

The HTML element selected by the expert constitutes a positive sample, whereas all the other HTML elements on the same web page constitute negative sample set. Web pages in our archive contain approximately 1000 to 3000 HTML elements each. The system forms feature vector based on the position information of elements. It then trains binary classifiers for each element which will take place in the new design. For example, if the expert wants to retarget 10 elements, 10 separate binary classifiers are trained and saved.

To obtain additional positive samples, the system acquires three other random days' data to supplement the test set, makes predictions and shows predicted elements to the user. At this point, the user can either approve or reject the prediction. If a prediction is approved, the new positive sample is used to improve the classifier. If it is rejected, the expert is expected to make another selection on the new page.

This is not an easy machine learning problem because the dataset is quite unbalanced. It has a very small number of positive samples (1 to 4) and relatively large number of negative samples. Since we used position based features and very few positive samples, we used the KNN algorithm.

3.5. Translation Rule Library

During translation, each action of the expert is assigned to a translation rule. Moving an image or a block of text to the adapted page or changing the color of the adapted page can be seen as the examples of different translation rules. The translation rule library consists of individual translation rule classes; one of each accomplishes a different task.

The architecture of translation rules is shown in Figure 3-6. All translation rules are extended from a base class. Each translation class implements design methods. The main working principle of the design methods is to find the correct place in the HTML/CSS code of the adapted page and modify it. When an instance of a translation rule class is generated, either by the multimodal user interface or the translation engine, design methods run and apply necessary changes to the adapted web pages.

Translation rule instances which are generated by the user interface, along with their properties, must be saved in such a way that the detail is preserved and the translation engine can execute them properly. The system achieves this using serialization/ deserialization processes. Translation model is formed as the final step of the translation process.

There are 7 translation rule classes that we defined:

- Rename tab page: Changes the name of a tab page.
- Color change: Changes the color of a page.
- Background color change: Changes the background color of a page.
- Copy object: Takes an element id and places this element to the adapted page.
- Copy logo: Copies the logo element to the top-left position of a page
- Change font: Changes the font size.
- Add tab page: Adds a new tab page to tab set on the page.

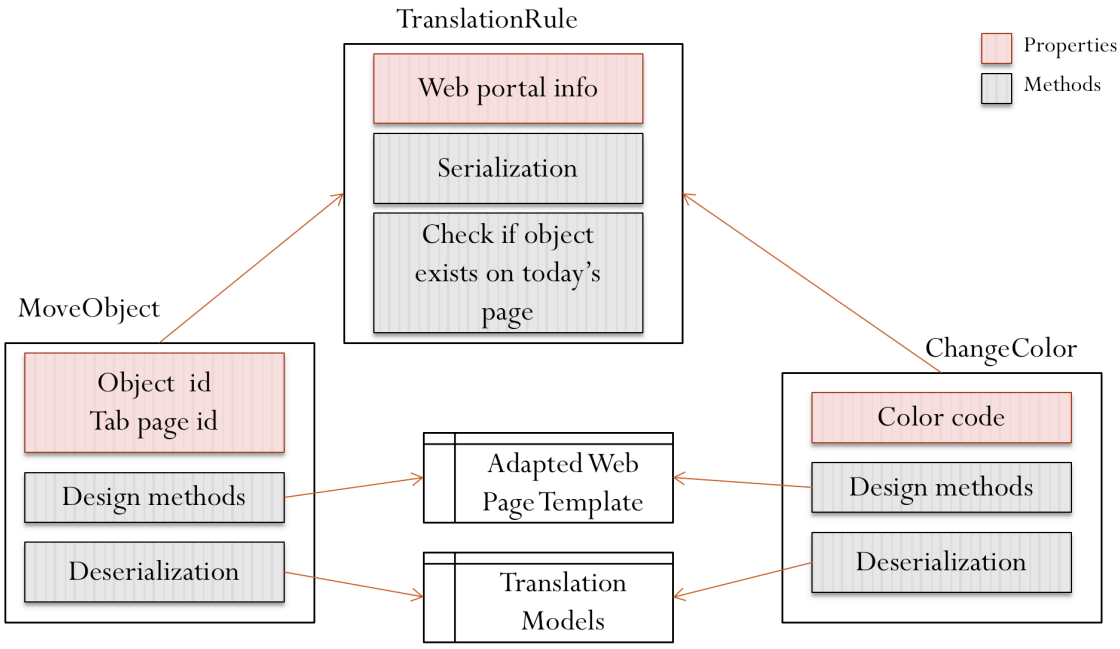


Figure 3-6: The architecture of translation rules

3.6. Translation Engine

The translation engine is the part that extracts pre-defined translation models, executes translation rules and creates adapted pages. If any serialized translation model exists for a web site, translation engine first de-serializes it. De-serialized translation model file contains translation rule instances. For example, if one of the translation rules is to move an image, resulting file will contain information about html element id of this image and classifier id to make a prediction for this image. When the translation rule runs with these parameters, its design methods will add html code of this image to the adapted version of web page.

After running all translation rule instances, the code of adapted web page is formed and saved. The translation engine then renders and views this new web page. Figure 3-7 shows an adapted version of www.reuters.com web page.

Figure 3-7: An adapted web page created by the translation engine.

3.7. Sample Translations

In Figure 3-8, we present samples of adapted versions of another web page, namely www.recipe.com.

Figure 3-8: Adapted versions of recipe.com. Two pages are obtained by applying the same translation model to different days' data.

Chapter 4

MULTIMODAL INTERACTION DESIGN

Multimodal interaction offers users a more natural and powerful way of communication via speech, touch, gestures, pen, eye gaze, and other modalities. Multimodal interaction research aims to develop new interaction technologies and interfaces which suffer less from the existing constraints and exploit more of what is possible in human-computer interaction [12]. The user interface that we implemented uses speech and sketch modalities to increase the usability of the system and to provide a powerful means of communication.

4.1. Speech Modality

Speech is one of the most prevalent modalities in recent multimodal systems thanks to its intuitive nature. IPhone's Siri and Google's Voice Search are famous examples of speech based applications. With IPhone's Siri, users can control their phones by voice, and with Google's Voice Search, users can perform an Internet search verbally. Similarly, users can operate our system via speech, and the system gives a voice response to users.

We use Microsoft Speech API (SAPI) [16] to implement a speech recognition engine. SAPI is easy to implement and achieves high recognition accuracy. Grammar files are used to specify the words and patterns of words to be identified by the recognition engine. We defined a grammar file in XML format complying with the Speech Recognition Grammar Specification (SRGS) [17]. We present a part of the grammar file that we created in Figure 4-1. This grammar block recognizes a sentence pattern which users can say to

move an HTML element. The rule implies that the sentence should be in the following format:

— Starts with either ‘copy’ or ‘move’

— Followed by another pattern defined by the sub-rule ‘Webel’

— Followed by ‘to’

— Followed by an optional ‘the’

— Ends with another pattern defined by the sub-rule ‘Position’

Sub-rule ‘Webel’ recognizes a word if it is one of the words ‘object’, ‘element’ and ‘item’. An example sentence which is recognized by this grammar would be “Copy element to tab page two”. The grammar file contains many patterns like this example which cover each translation rule and their verbalizations in multiple ways.

```
<rule id="Copy2" scope="public">
  <one-of>
    <item> Copy </item>
    <item> Move </item>
  </one-of>
  <item> <ruleref uri="#Webel"/> </item>
  <item> to </item>
  <item repeat="0-1"> the </item>
  <item> <ruleref uri="#Position"/> </item>
</rule>

<rule id="Webel" scope="public">
  <one-of>
    <item> object </item>
    <item> element </item>
    <item> item </item>
  </one-of>
</rule>

<rule id="Position" scope="public">
  <one-of>
    <item> tab page <ruleref uri="#Number"/> </item>
    <item> tab <ruleref uri="#Number"/> </item>
    <item> current tab page </item>
    <item> active tab page</item>
  </one-of>
</rule>

<rule id="Number">
  <one-of>
    <item> one <tag> out = 1; </tag> </item>
    <item> two <tag> out = 2; </tag> </item>
    <item> three <tag> out = 3; </tag> </item>
    <item> four <tag> out = 4; </tag> </item>
  </one-of>
</rule>
```

Figure 4-1: A part of the grammar file

4.2. Sketch Modality

Sketching is yet another important modality as it offers the flexibility of paper and pencil use on digital devices. To provide this flexibility to users, we designed a drawable user interface. Sketching enables users to perform some operations that would require more efforts through traditional modalities. When using the system, users select HTML elements on a web page. One way of doing this might be highlighting each possible element on hover interactively. However, this is elaborate and tedious for users since there are many possible HTML elements for each hover. Performing this task by sketching makes it simpler and more elegant. Similarly, users can move an HTML element to the adapted design by drawing an arrow sketch on screen. We implemented an algorithm which detects sketches and sketch points, differentiates between the corresponding tasks, and transfers the control to the necessary part of the program.

4.3. Interaction Principles

To improve the quality of the multimodal user interface and to provide better communication between the system and users, we added three features to the system.

One of the most important principles of the user interface design is the feedback principle. The feedback principle states that the design should inform the users about actions or interpretations, state changes and errors through an explicit and clear language [20]. To achieve this, we built a hierarchical feedback mechanism consisting of 3 feedback categories. The first category is ‘acknowledgment’ and this kind of feedback is given to the user if the state of the system is changed, but the user does not need to take any action about it. A text message appears at the bottom of the screen in such cases without interrupting the user. The second category is ‘information’ and this kind of feedback is given to users to inform them about a state change due to their actions, i.e. the system took the action that they demanded. Speech synthesis is used for this type of feedback without

interrupting the flow, i.e. a voice tells the message to the user. The last category is called ‘error’. In error cases, a pop-up message appears on the screen to interrupt users and make them correct the mistake before continuing. The error messages are designed in a non-technical and intuitive way so that users can fully grasp the reason for the error and can fix it.

The second feature we added to improve the interaction works in the following way. After learning an HTML element, the system makes predictions on three other pages and shows them to the user. We added this feature for two main reasons. The first reason is to convince the user that the system does not memorize this element, instead, it learns it in order to be able to predict it on different pages. Second, if the prediction is wrong, the system acquires more positive samples after the expert’s new feedback on these new pages.

As the third feature, we used audio cues, colors and highlights to help users to engage with the system more. Using these features also satisfies the simplicity principle which states that the interfaces should be implemented in the simplest possible yet an engaging way.

Chapter 5

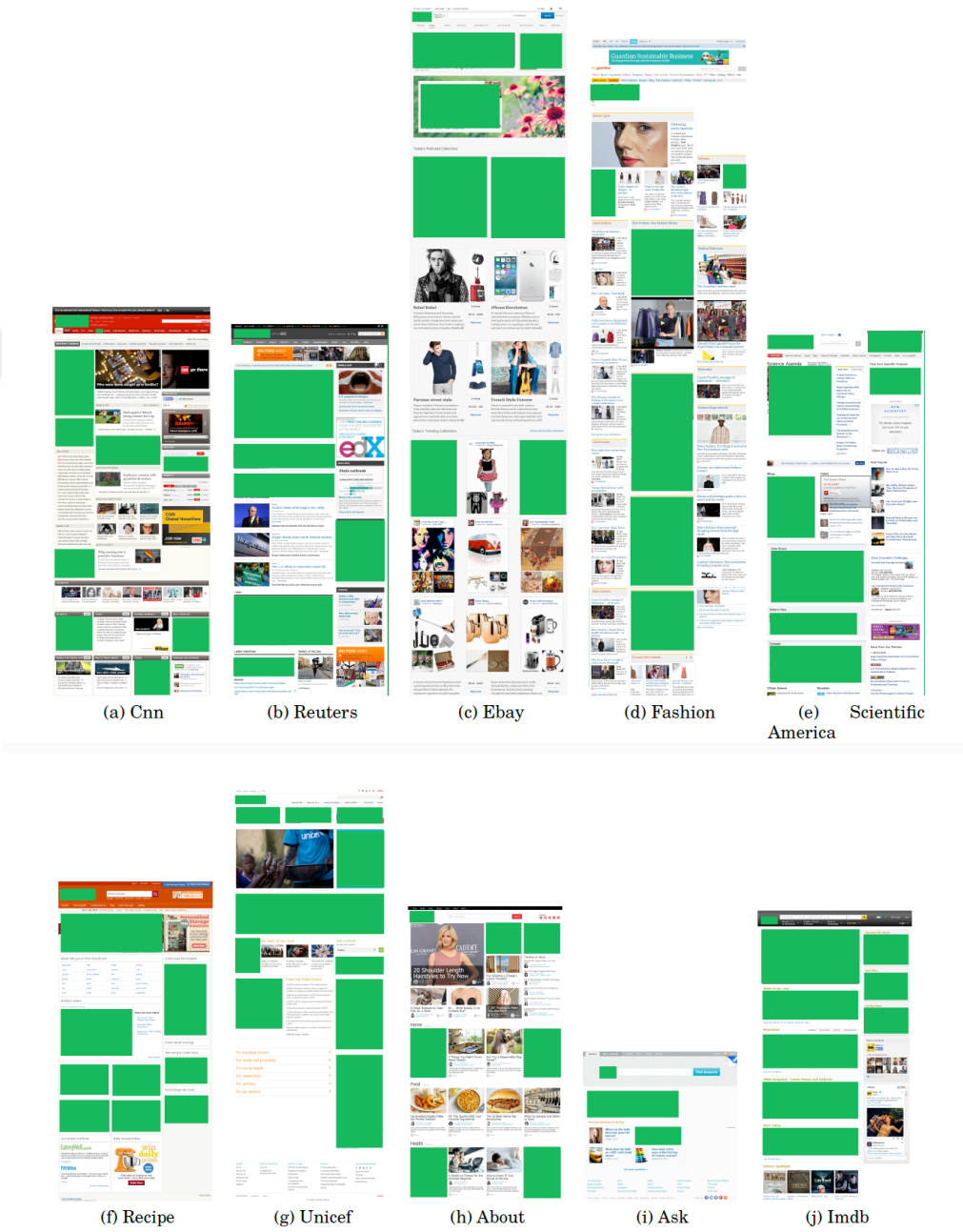
EVALUATION OF WEB ELEMENT LEARNING

When the system builds an adapted web page based on a translation model, all the translation rules except for moving an HTML element are guaranteed to modify the page correctly. On the other hand, whether or not to place an HTML element to a new design depends on the correct prediction of that element. Hence, the precision of an adapted page is directly correlated with the prediction accuracy of HTML elements. To assess the performance of the system in terms of thoroughly forming the adapted pages, we evaluated the web element learning process.

5.1. Experiment Design

We conducted our experiments on the websites from the web archive. Since we used location-based features, the more an HTML element is in the same place and the same size, the less prediction error there is. For example, the logo of a website is generally easy to predict because it is in the exact same place and in the same size every day. However, an article at the bottom of a news website is harder to predict because the content divisions of a news website are unstable due to the varying content amount. Moreover, the location of each content division is effected by the content above.

We defined a number of objects from different difficulty levels for each website. We implemented an annotation tool and annotated these objects on all the pages in the archive, and created binary feature vectors for each object. Figure 5-1 shows the objects we annotated for each website.



(a) Cnn (b) Reuters (c) Ebay (d) Fashion (e) Scientific America

(f) Recipe (g) Unicef (h) About (i) Ask (j) Imdb

Figure 5-1: Objects we annotated for each website.

The system trains classifiers for HTML elements using very few day's data and makes predictions using these classifiers on an unseen day's data. We created the same setup for the experiments in the following way.

We first conducted the experiments by using one positive sample for each object. Then we repeated the experiments for each object twice by incrementing the positive sample count by one each time. In other words, we respectively used 1, 2 and 3 days' data to retrieve the positive samples and tested the system with one of the unused days each time to calculate the accuracy on unseen data. For example, for cnn.com which has 30 pages in the archive, we conducted the experiments for:

- $\binom{30}{1}$ times training, and for each training set, the number of unused day times testing (30*29 test)
- $\binom{30}{2}$ times training, and for each training set, the number of unused day times testing (435*28 test)
- $\binom{30}{3}$ times training, and for each training set, the number of unused day times testing (4060*27 test)

which adds up to total 870+12180+109620 runs. Figure 5-2 shows our evaluation algorithm.

Even though the algorithm predicts more than one element as positive, we have to pick one of them since we must place only one element to the adapted page. Therefore, if the system predicts more than one positive value, we picked one of them as positive and converted the other's labels into negative. We called this process "unification". During the HTML element matching phase from the expert's drawings, the algorithm picks the element which has the longest HTML code among the best matches. Similarly, during unification, we picked the element which had the longest HTML code.

Since each test file contains one positive sample and a massive number of negative samples, the conventional accuracy becomes non-informative. In this classification

problem, the success of the system is binary, either the system predicts the positive sample correctly, or not. Thus, instead of calculating accuracy, we calculated true positive, true negative, false positive and false negative numbers in each run. There are two possible cases: 1) If the false negative number is 1 (i.e. true positive number is 0), it means the system failed. 2) If the false negative number is 0 (i.e. true positive number is 1), it means the system either predicted only the actual positive sample as positive, or there might be additional false positives. In this case, the system applies unification and updates TP, TN, FP and FN numbers. Considering the system applies unification, the true positive rate indicates the accuracy of the system. We run the experiments for each object and for each possible training/test combination and reported the average true positive counts as the accuracy of the system.

Algorithm 1 Evaluation Script

```
1:  procedure EVALUATION
        Parameters
2:      website ← string website name
3:      element_id ← integer element id
4:      positive_count ← integer positive training day count
5:      features ← [1 1 1 1]

        Steps
6:      for i=1..(daycount choose positive_count) do
7:          Xtrain, ytrain ← feature,label matrix with included features
8:          train_indicator[daycount] ← mark training days
9:          for j=1..daycount do
10:             if train_indicator[j] is off then
11:                 Xtest, ytest ← feature,label matrix with included features
12:                 run KNN algorithm
13:                 accuracy, err_cnt ← updated average values
14:     avg_acc, avg_errcnt ← avg_acc, avg_errcnt.
```

Figure 5-2: Evaluation algorithm

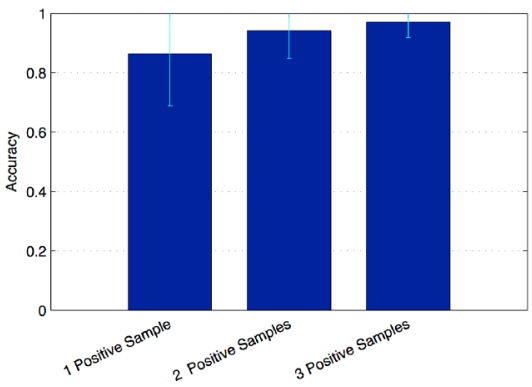
5.2. Results

We present the averaged accuracies for all objects in Table 5-1 and the resulting graphs in Figure 5-3.

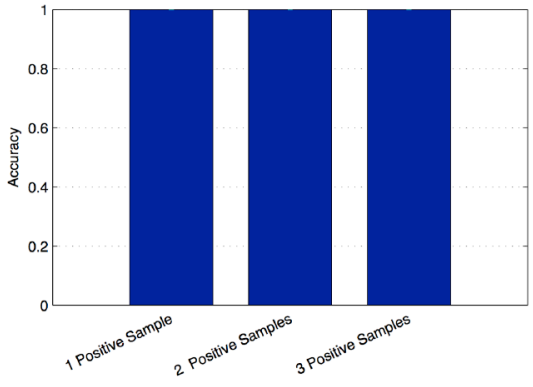
Websites with fixed content divisions like ask.com and unicef.org achieve high accuracies. On the other hand, websites which do not have stable content like cnn.com and reuters.com might achieve relatively lower accuracies. The overall accuracies increase as the training day count increases. These results show that even if a website contains hard-to-predict objects, overall accuracies are still successful. Hence our system can learn web elements.

Table 5-1: Average accuracies for websites

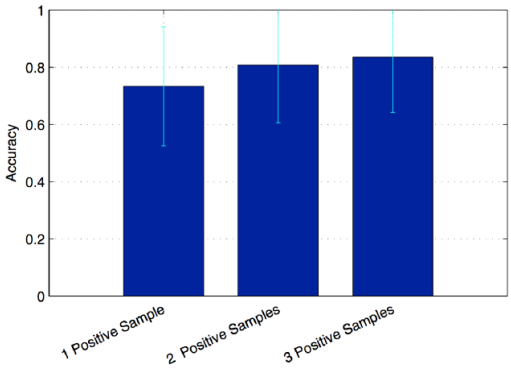
	1-day acc	2-day acc	3-day acc
About.com	0.88	0.96	0.98
Ask.com	1	1	1
Cnn.com	0.73	0.81	0.84
Ebay.com	0.96	0.98	0.98
Fashion.com	0.93	0.97	0.98
Imdb.com	0.79	0.85	0.89
Recipe.com	0.69	0.83	0.87
Reuters.com	0.78	0.83	0.86
ScientificAmerica.com	0.88	0.90	0.91
Unicef.org	1	1	1
Average	**0.86**	**0.91**	**0.93**



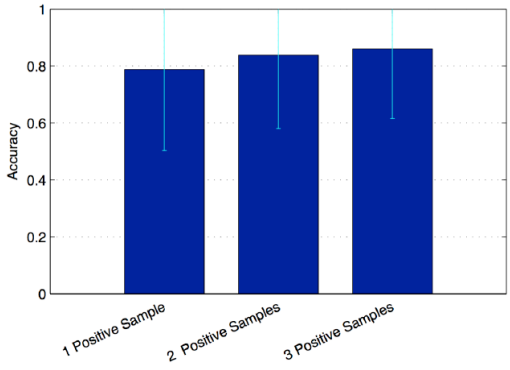
a) ask.com



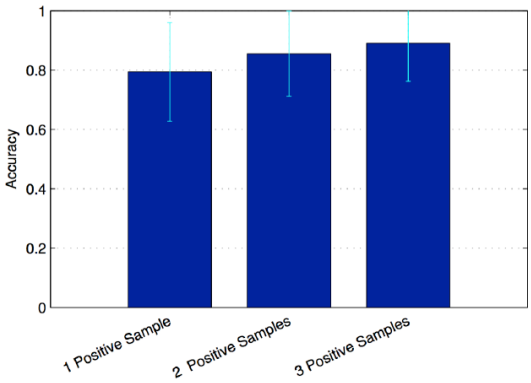
b) about.com



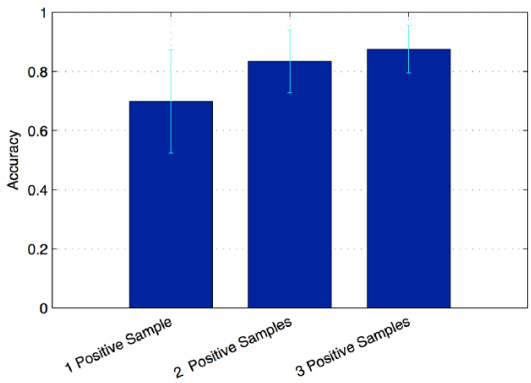
c) cnn.com



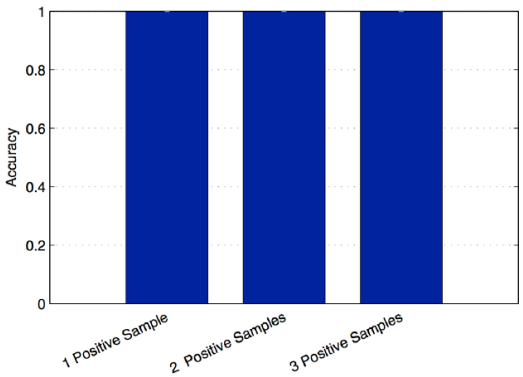
d) reuters.com



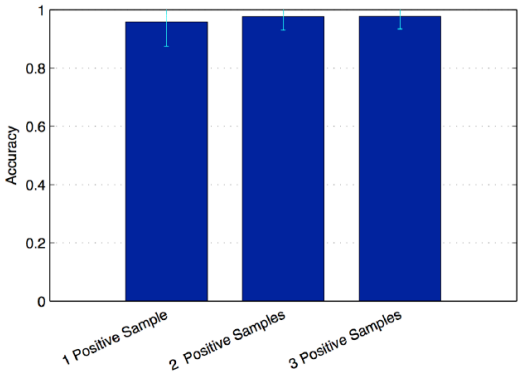
e) imdb.com



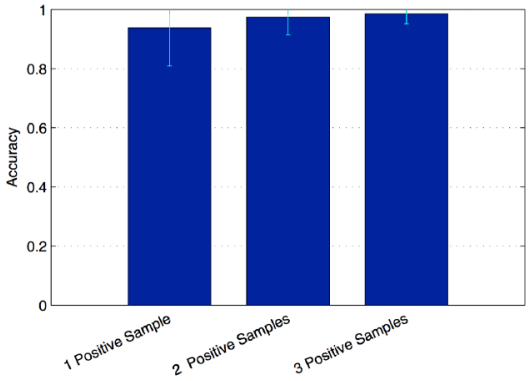
f) recipe.com



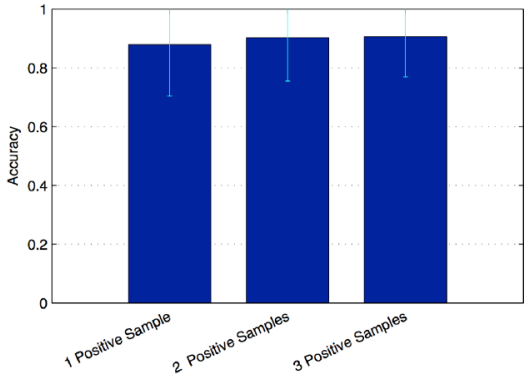
g) unicef.com



h) ebay.com



i) theguardian.com/fashion



j) scientificamerican.com

Figure 5-3: Average accuracy graphs

Chapter 6

USABILITY STUDY

We empirically evaluated the usability of the system by testing it with real users. The purpose of this test is to assess whether real users are able to learn how to use the system and they can create translation models by using it. We collected subjective and objective measurements to evaluate the system.

We ran the experiments on a setup composed of a Wacom Cintiq 24 HD Touch tablet with a pen stylus and a computer with Intel(R) Core(TM) i7, 3.07 GHz processor and 8GB RAM. Figure 6-1 shows the experimental setup.

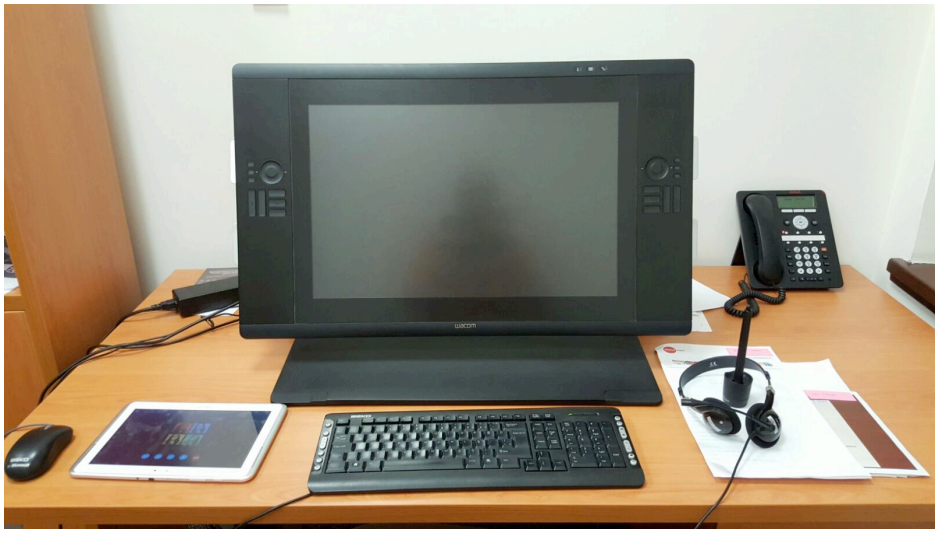


Figure 6-1: Environmental Setup

6.1. Pilot Study

Prior to the usability study, we conducted pilot studies with three people. We asked them to use the system and observed them while they were using the system. The aim of these studies was to observe the performances of the subjects and make the necessary improvements and changes in our system.

6.2. Design of the Study

We divided a session into three parts: the training part, the main part, and the questionnaire part.

The first part of the study was the training part in which we taught users how to use the system. In this part, users were first asked to watch a video showing a user who uses the system. Watching this video gives users a quick and accurate understanding of the system. Following the video, the users were asked to self-study a presentation slide which explains how to use the system. Afterwards, they first practiced the speech interaction by performing a translation task. We informed them about that was a practice and they should focus on using speech modality. Subsequently, they were asked to perform one more practice study on a different web portal. They were informed that this was also a practice and the purpose was to give them an opportunity to create an adapted page. During these practice studies, they were allowed to ask questions. Afterwards, they were told that the first part of the study was over and the second part, namely the main part, was beginning. In the main part, users were not allowed to ask any questions. We had created designs for different web portals by using our system and printed them out. Users were given these printed versions of adapted designs and were asked to create the exact same design. While they were using the system, we observed them and noted how many errors they made, how well they recovered from these errors and what percentage of the design they created. We used this information to evaluate the usability of the system objectively.

Finally, the third part of the usability study was to fill out a questionnaire. After the second part was completed, we asked users to fill out a questionnaire about the system. We used the results of these questionnaires to evaluate the usability of system subjectively.

6.3. Experiments

10 people (6 males and 4 females) between the ages of 19 and 28 participated in our usability study. Participation was voluntarily and the participants were free to leave anytime. Each session lasted approximately 60 minutes. Figure 6-2 shows a picture from the usability study of a user.

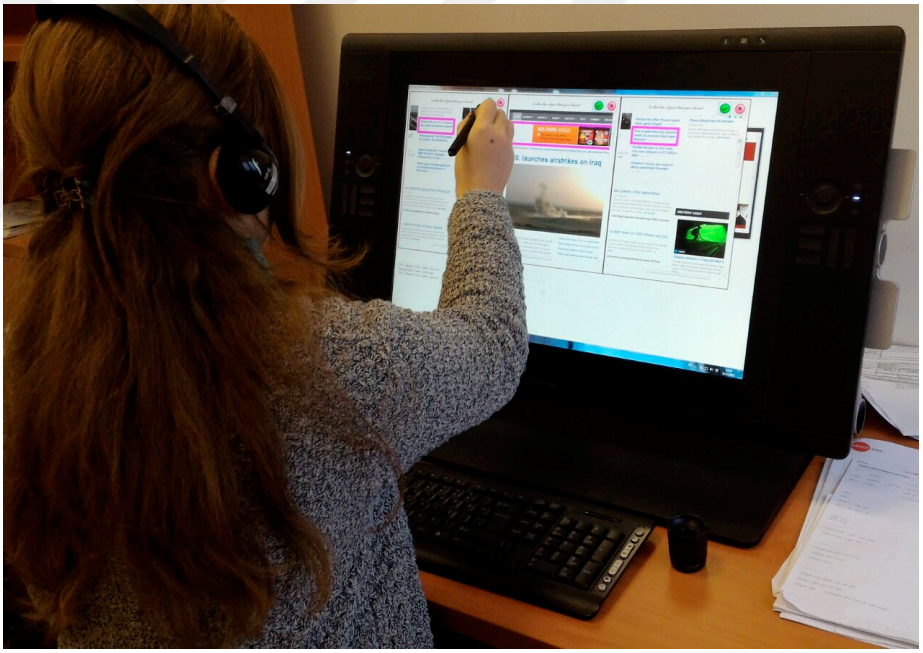


Figure 6-2: A participant is using the system during the usability study

6.4. Questionnaire

We designed a questionnaire for the subjective evaluation of the system which contains 9 open-ended questions asking information about the participants such as their age or previous experience on multimodal user interfaces for statistical purposes and 22 questions about the system. We defined 9 categories and created negative and positive forms of questions for each category (22 questions in total). The negative and positive questions were distributed randomly through the questionnaire so that none of the questions belonging to the same category followed each other. Table 3 shows the categories and the number of questions belonging to each category.

Table 6-1: Question categories for questionnaire

Category	Category Name	Question number in this category
1	Pleasant to Use	2
2	Error Recovery	2
3	Task Accomplishment	3
4	Enjoy Interaction	3
5	Enjoy Speech Usage	3
6	Enjoy Sketch Usage	2
7	Multimodal vs. Manual	3
8	Easy to Use	2
9	Understand How to Use	2

6.5. Subjective Results

We calculated the subjective results based on the questionnaires filled by the participants. Each user was asked 22 questions from 9 categories. We used 7-point Likert scale and the levels were ‘Strongly Disagree’, ‘Disagree’, ‘Somewhat Disagree’, ‘Neither Agree Nor Disagree’, ‘Somewhat Agree’, ‘Agree’, ‘Strongly Agree’. Answers to the negative and positive questions of the participants were consistent with each other within a single point scale interval. To be able to aggregate the scores of all questions in the same category, we first calculated the complementary scores for the negative questions. Then we computed the average scores for each category with all users’ data. Figure 11 shows the average results of 10 users for each category. All the categories have a score of higher than 6 and the average is 6.3. The numbers suggest that users enjoyed using our system, they could learn how to use the system, they could build translation models by using it and they enjoyed interacting with the system.

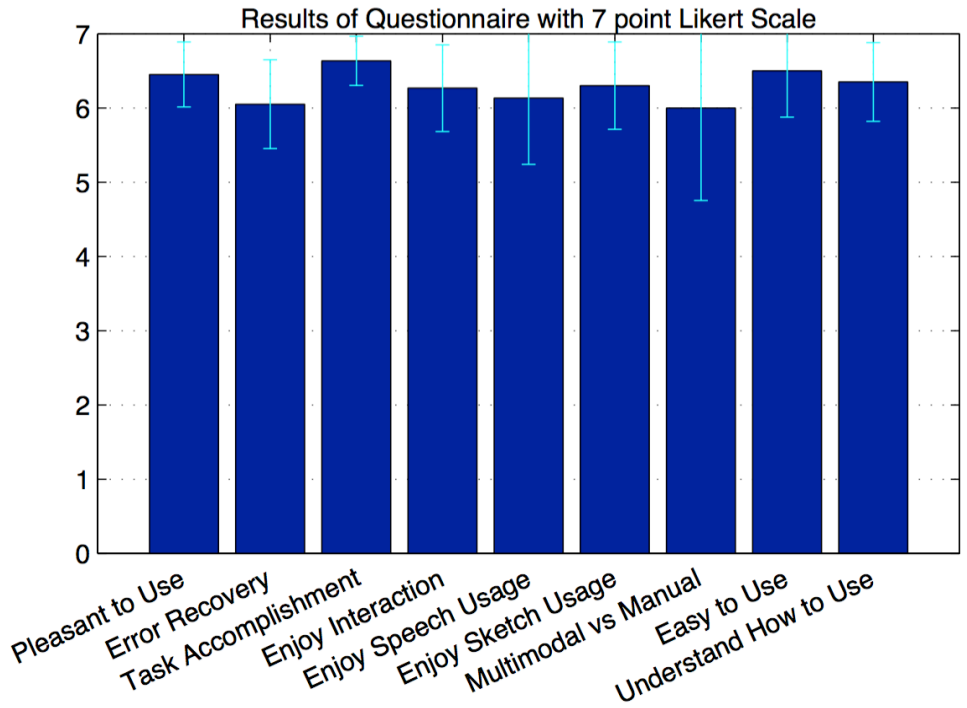


Figure 6-3: Average results for each category. We averaged the results for all questions in each category and for all users.

At the beginning of the questionnaire, we asked our users about their previous experiences on speech based or sketch based interfaces. Four out of ten users explicitly stated that they experience speech interaction in their daily lives via different technologies. We called this group as "tech-savvy" group. Three out of ten users stated that they are 'shy' about using speech commands and they avoid using it. We called this group as "tech-averse" group. The remaining three users who gave neutral answers were omitted. We compared the answers of the tech-averse group and the tech-savvy group for two categories of questions: 1) Multimodal interaction based questions and 2) Usability based questions. Figure 12 shows the resulting graphs. For the multimodal interaction based question group,

the average score of the tech-savvy group is 6.66 points, whereas the average score of the tech-averse group is 5.36 points. On the other hand, for the usability based question group, the average score of the tech-savvy group is 6.61 points, whereas the average score of the tech-averse group is 6.2 points which is much closer to the tech-savvy group. These results show that even tech-averse users can use our system successfully and more importantly, they enjoy using the system.

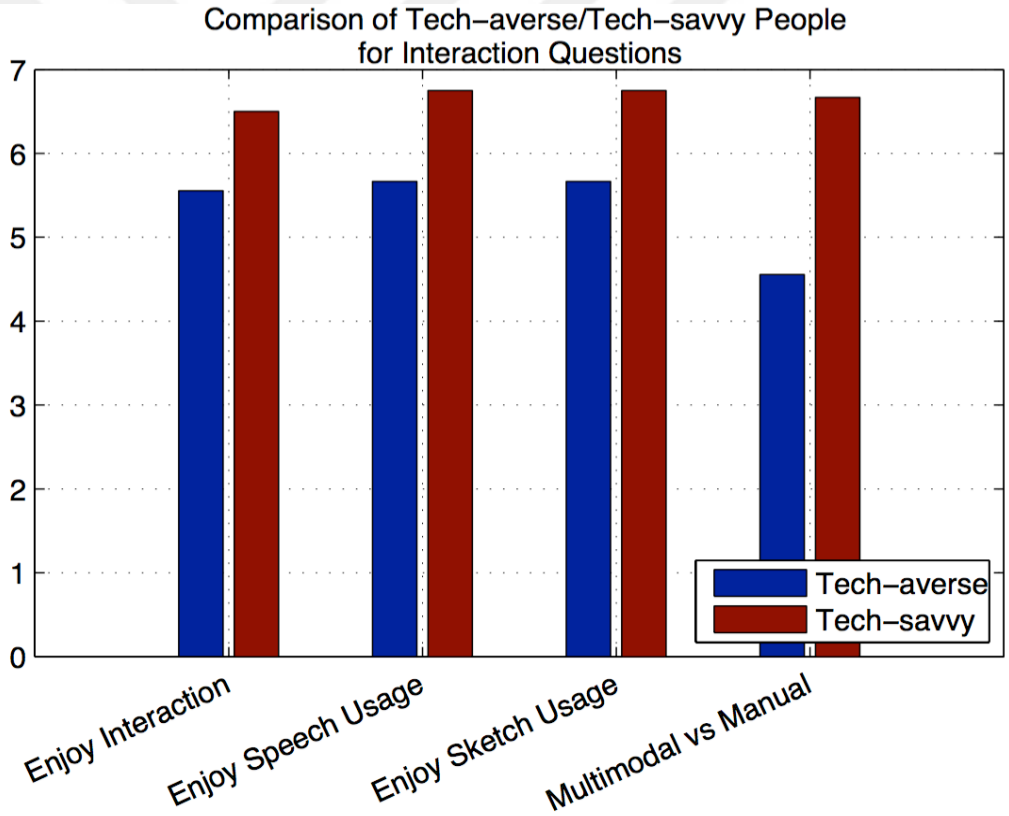


Figure 6-4: Comparison of the answers of tech-averse and tech-savvy users for interaction questions.

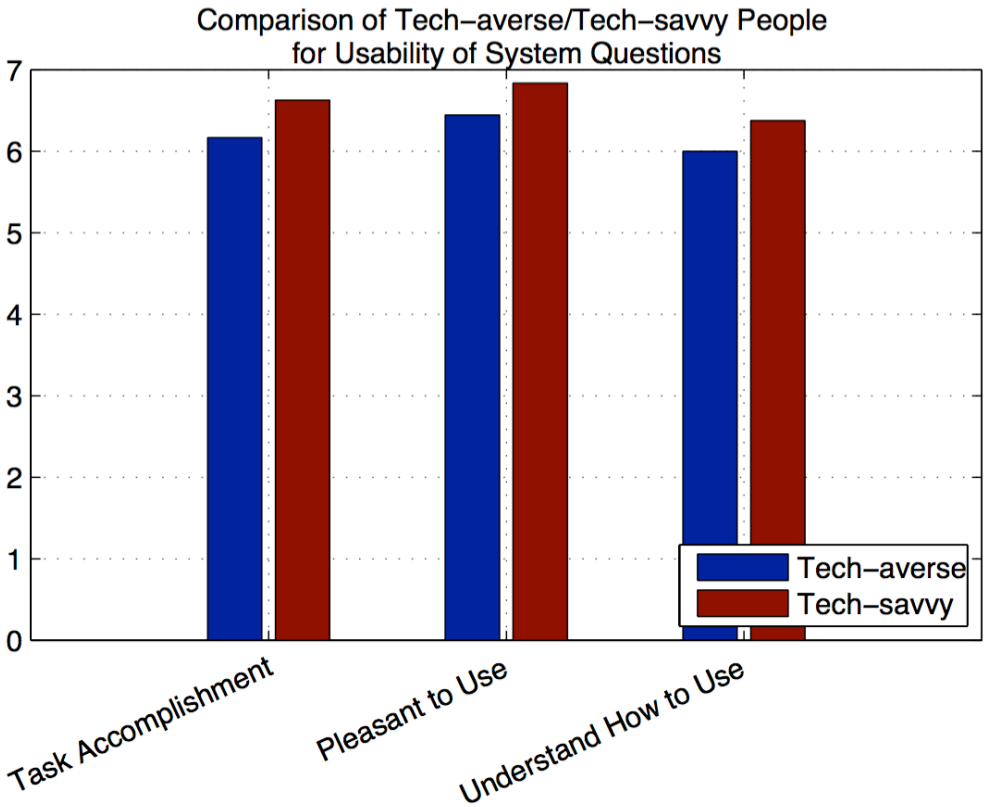


Figure 6-5: Comparison of the answers of tech-averse and tech-savvy users for system usability questions.

6.6. Objective Results

Other than the subjective evaluation, we also assess the usability of the system objectively by using task accomplishment and error rates. We measured the task accomplishment rate in the following way. In the main part of the experiments, users are expected to replicate the designs of two web portals. We only showed users the completed designs and they tried to create the same design on MICA. When building the designs in the first place, we used a number of translation rules. We measured what percentage of the initial designs users could be able to replicate.

During the study, we observed the users and counted the errors made by each user and noted whether they could recover from those errors or not. There are two types of errors: user errors and system errors. If users do something logically wrong, this is a called a user error. Sometimes, the system misinterprets the user and takes an action. This is called a system error. We designed error messages to help users to recover from both kinds of errors. An example of a system error is:

— User coughs

— System interprets the cough as “Move” command erroneously and moves an element

An example of a user error is:

— User tries to move an object to a non-existing tab page or tries to move an object without selecting any objects

In these types of situations, users successfully recovered from the error. Figure 13 shows the task accomplishment and error rates for each user. The results show that users successfully operated the system and they could recover from the errors.

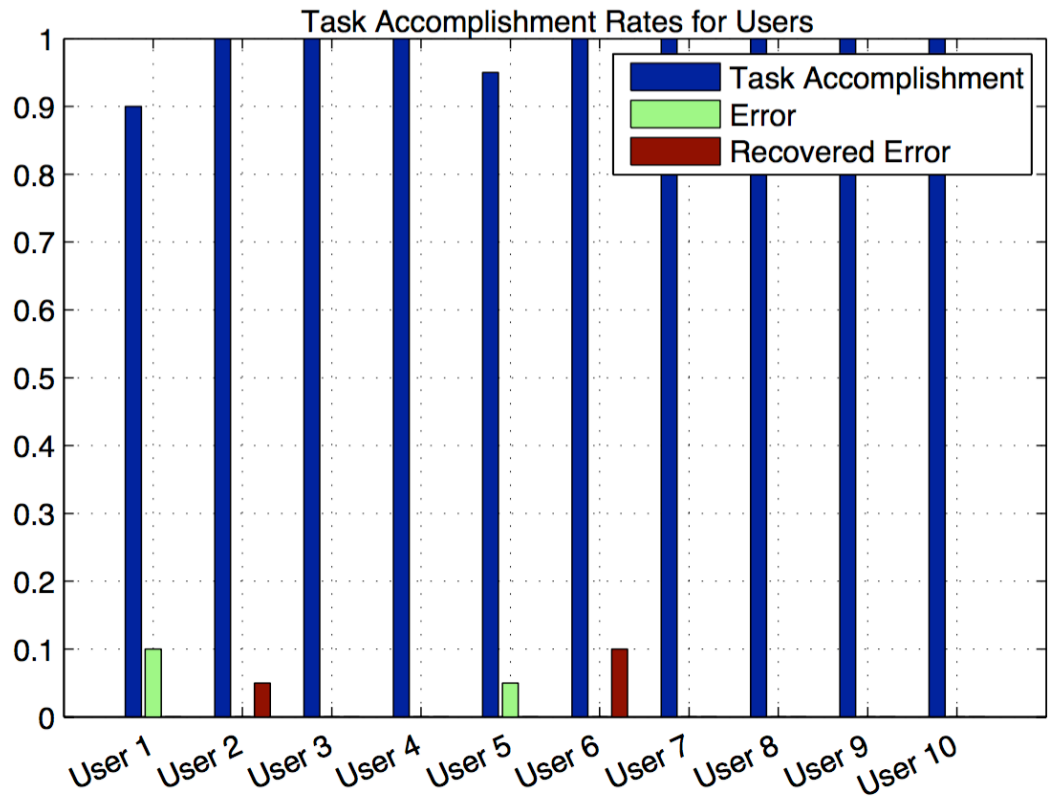


Figure 6-6: Task accomplishment rates fro each user

Chapter 7

CONCLUSION AND FUTURE WORK

In this paper, we introduced a system architecture and implementation of a semi-automatic multimodal content translation system to create new designs tailored to a specific device for an existing website. Our system consists of two independent components. The first component learns translation rules from few examples using machine learning and human expertise. The second component executes the translation models and builds adapted versions of previously unseen web pages.

We evaluated the system in terms of learning of translation rules and the usability of the system. The results of learning evaluation show that the system can learn HTML elements and correctly predict these elements on unseen pages with an average 93% accuracy. The results of usability studies show that real users can learn how to use the system and can create translation models by using it.

This system can be used in various different ways and by different subjects including:

- Next generation device manufacturers: They can create custom-fit designs of web pages for their devices and let their users to view these designs instead of original web pages.
- Individual users: Individual users can personalize web pages by using our system. For example, if an economist visits a web page on a daily basis to check a certain information, he can personalize this website and create a simpler version of it.

- Parental guidance: Parents can use this system to eliminate some parts of the pages and let their children view the new design.
- Accessibility for the people with disabilities: Custom designs can be developed for disabled people. For example, a simpler version of a website with bigger fonts can be created for a person with a limited vision.

The framework we built is extendable. One of our future steps is to add new translation rules to the system so that it can create more advanced web pages. We also want to improve the features that we used for web element learning. We have many different kinds of data including text and image information of thousands of web elements. We plan to run different experiments with this data to increase the accuracy. Another future work is to create common translation models for groups of websites. We are curious and excited about further exploration of our system and the dataset.

BIBLIOGRAPHY

[1] Hwang, Yonghyun, Jihong Kim, and Eunkyong Seo. "Structure-aware web transcoding for mobile devices." Internet Computing, IEEE 7.5 (2003): 14-21.

[2] Chen, Yu, et al. "Adapting web pages for small-screen devices." Internet Computing, IEEE 9.1 (2005): 50-56.

[3] Lee, Eunshil, et al. "Topic-specific web content adaptation to mobile devices." Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE Computer Society, 2006.

[4] Cai, Deng, et al. Vips: a vision-based page segmentation algorithm. Microsoft technical report, MSR-TR-2003-79, 2003.

[5] Anam, Rajibul, Chin Kuan Ho, and Tek Yong Lim. "Web content adaptation for mobile devices: A greedy approach." International Journal of New Computer Architectures and their Applications (IJNCAA) 1.4 (2011): 1027-1042.

[6] Xiao, Xiangye, et al. "Browsing on small displays by transforming Web pages into hierarchically structured subpages." ACM Transactions on the Web (TWEB) 3.1 (2009): 4.

[7] Chen, Yu, Wei-Ying Ma, and Hong-Jiang Zhang. "Detecting web page structure for adaptive viewing on small form factor devices." Proceedings of the 12th international conference on World Wide Web. ACM, 2003.

[8] Adzic, Velibor, Hari Kalva, and Borko Furht. "A survey of multimedia content adaptation for mobile devices." Multimedia Tools and Applications 51.1 (2011): 379-396.

[9] Nebeling, Michael, Maximilian Speicher, and Moira C. Norrie. "CrowdAdapt: enabling crowdsourced web page adaptation for individual viewing conditions and preferences." Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems. ACM, 2013.

[10] Adipat, Boonlit, Dongsong Zhang, and Lina Zhou. "The effects of tree-view based presentation adaptation on mobile web browsing." Mis Quarterly 35.1 (2011): 99-122.

[11] Wei, Chenjie, et al. "Assisted human-in-the-loop adaptation of Web pages for mobile devices." Computer Software and Applications Conference (COMPSAC), 2013 IEEE 37th Annual. IEEE, 2013.

[12] Turk, Matthew. "Multimodal interaction: A review." Pattern Recognition Letters 36 (2014): 189-195.

[13] Schalkwyk, Johan, et al. "“Your Word is my Command”: Google Search by Voice: A Case Study." Advances in Speech Recognition. Springer US, 2010. 61-90.

[14] Blessing, Alexander, et al. "A multimodal interface for road design." International Conference on Intelligent User Interfaces Workshop on Sketch Recognition. 2009.

[15] Rossetto, Luca, et al. "IMOTION—A Content-Based Video Retrieval Engine." MultiMedia Modeling. Springer International Publishing, 2015.

[16] MSDN. 2016. Microsoft Speech API (SAPI) 5.4. https://msdn.microsoft.com/en-us/library/ee125663. (2016). [Online; accessed 19-March-2016].

[17] World Wide Web Consortium. 2016. Speech Recognition Grammar Specification Version 1.0. https://www.w3.org/TR/speech-grammar/. (2016). [Online; accessed 19-March-2016].

[18] Hattori, Gen, et al. "Robust web page segmentation for mobile terminal using content-distances and page layout information." *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007.

[19] Hofmann, Hansjörg, et al. "Comparison of speech-based in-car hmi concepts in a driving simulation study." *Proceedings of the 19th international conference on Intelligent User Interfaces*. ACM, 2014.

[20] Constantine, Larry L., and Lucy AD Lockwood. *Software for use: a practical guide to the models and methods of usage-centered design*. Pearson Education, 1999.