Chapter 12
Prototypes and Demonstrators

With Contribution Of: Luis A. Leiva, Vicent Alabau, Verónica Romero, Franco M. Segarra, Ricardo Sánchez-Sáez, Daniel Ortiz-Martínez, Luis Rodríguez, and Nicolás Serrano.

Contents

12.1 Introduction 228
12.2 MM-IHT: Multimodal Interactive Handwritten Transcription 231
12.3 IST: Interactive Speech Transcription 239
12.4 IMT: Interactive Machine Translation 242
12.5 ITG: Interactive Text Generation 246
12.6 MM-IP: Multimodal Interactive Parsing 251
12.7 GIDOC: GIMP-Based Interactive Document Transcription 255
12.8 RISE: Relevant Image Search Engine 261
12.9 Conclusions 264
References 265

This chapter presents several full working prototypes and demonstrators of multimodal interactive pattern recognition applications. These systems serve as validating examples of the approaches that have been proposed and described throughout this book. Among other interesting things, they are designed to enable a true human–computer interaction on selected tasks.

To begin, we shall expound the different protocols that were tested, namely Passive Left-to-Right, Passive Desultory, and Active. The overview of each demonstrator is sufficiently detailed to give the reader an overview of the underlying technologies. The prototypes covered in this chapter are related to transcription of text images (IHT, GIDOC), machine translation (IMT), speech transcription (IST), text generation (ITG), and image retrieval (RISE). Additionally, most of these prototypes shall present evaluation measures about the amount of user effort reduction at the end of the process. Finally, some of such demonstrators come with web-based versions, whose addresses are included to allow the reader to test and practice with the different implemented applications.

12.1 Introduction

Throughout this book, the multimodal interactive–predictive paradigm to pattern recognition (MIPR) has been theoretically studied. The proposed paradigm has been empirically validated on controlled laboratory experiments with computer-simulated users. Although this set-up provides a reasonable framework for research, it is relatively optimistic regarding user interaction when dealing with real applications.

In this chapter we present some implementations of prototypes for several MIPR tasks. They all entail a multimodal, interactive strategy, and they all fully integrate the user's knowledge into the Pattern Recognition (PR) process. These demo systems seek two goals: (1) allow potential users to have a quick view into the MIPR technologies, and (2) facilitate data acquisition to study real user interaction for evaluation purposes in a more realistic scenario. As we will expound in the next sections, the use of these prototypes may entail a drastic reduction of user effort without sacrificing usability—quite the opposite, since, as stated before, the Human–Computer Interaction (HCI) paradigm is considered the main actor in front of a PR scenario.

The organization of each section in this chapter is as follows. After a brief introduction, a description of the prototype's demonstration is presented. In each description, a user interaction protocol is both established and itemized. Then a section about the technologies involved in the creation of the prototype is introduced. Finally, a discussion about both the evaluation of the prototype (if available) and the main achieved results are delineated. In all user evaluations, there is a noticeable improvement over the traditional PR approaches. Thus, as commented in this chapter's preface, these demos serve as validating examples of the MIPR framework proposed and described throughout this book. User's feedback directly allows us to improve system accuracy, while multimodality increases system ergonomy and user acceptability. Multimodal interaction is approached in such a way that both the main and the feedback data streams help each other to optimize overall performance and usability. The prototypes presented below can be classified on the basis of the interaction protocols described in Sect. 1.4. Now we briefly introduce an outline of each prototype based on the above-mentioned schema.

12.1.1 Passive, Left-to-Right Protocol

The prototypes classified under this interaction protocol fulfill two requirements. On the one hand, this being a passive protocol, they are directed toward a full supervision, where a 'perfect' (high-quality) result is needed. On the other hand, in this interaction protocol a left-to-right order in the output constituents is assumed, which makes such protocol appropriate for human language processing tasks (notice that a right-to-left protocol is also perfectly applicable for those languages that require so, e.g., arabic or persian).

The passive, left-to-right protocol is defined by the next procedure: First, the system proposes a fully automatic output. Then, the user validates the longest prefix of the output which is error-free by entering some amendment keystrokes to correct the first error in the suffix. If the prototype supports multimodality, the interaction can be performed by different kinds of feedback channel, such as touchscreen pen strokes and/or other possible interaction modalities such as speech input. The system then suggests a suitable, new extended consolidated prefix based on the previous validated prefix, the multimodality decoding, and the keystroke amendments. These steps are repeated until the user is satisfied with the system's output.

The series of prototypes for natural language processing tasks that have been implemented following this protocol are introduced now.

Multimodal Interactive Handwritten Transcription (MM-IHT) Given a text image which corresponds to a line of a digitized handwritten document, the user transcribes the text in the image with the help of the system. Multimodality can be achieved by entering some pen strokes to amend the first error in the suffix.

Interactive Speech Transcription (IST) The user transcribes speech utterances from parliamentary proceedings, public speeches, video lectures, etc. Corrections are made by using the mouse and the keyboard.

Interactive Machine Translation (IMT) A document in a source language is loaded into the system. The application splits the document into sentences, which are then translated into a target language by using the keyboard, the mouse, and/or pen strokes.

Interactive Text Generation (ITG) The system works by predicting what the user is going to type based on the topic of the document, the context, and previously typed documents from the same user. Several input modalities can be used in order to adapt the system to different scenarios and/or user preferences.

Multimodal Interactive Parsing (MM-IP) The user generates interactively a syntactic analysis for an input sentence. One can perform modifications to the presented tree, using either the keyboard, the mouse, or other advanced input modalities.

As we shall see, most of these prototypes are built with the same communication toolkit [1], which offers an Application Programming Interface (API) that allows a TCP socket connection between client(s) and PR engine(s). Using sockets has several advantages over other communication channels. On the one hand it allows much faster message exchanging and lower latency responses. On the other hand, a multiuser environment can be easily implemented, so that several user across the globe can work concurrently on the same task. In addition, the web server and the PR engine do not need to be physically at the same place. Therefore, a dedicated PR engine can be run per task to deal with high CPU demanding corpora, or several web servers can be set up with the same task to serve an increasing amount of users.

Three basic functions summarize the above-mentioned API:

- `set_source`: selects the source phrase to be transcribed.
- `set_prefix`: sets the longest error-free prefix and amends the first error with keyboard strokes.

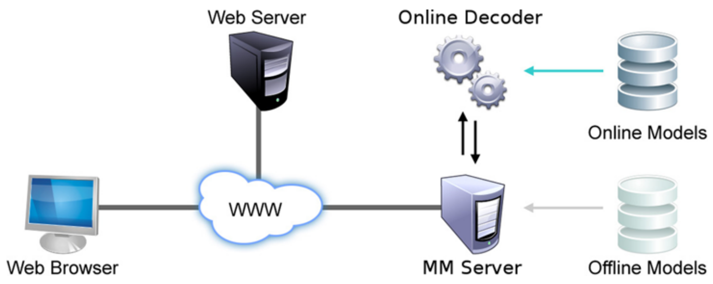


Fig. 12.1 Diagram of (common) system architecture for passive, left-to-right prototypes

- `set_prefix_online`: sets the longest error-free prefix and amends the first error with pen strokes.

Additionally, such prototypes share a common architecture, which is schematized in Fig. 12.1. The online form of such an architecture allows to carry out collaborative tasks with thousands of users across the globe, thus notably reducing the overall PR process. Since the users operate within a web browser window, the system also provides cross-platform compatibility and requires no disk space on the client machine.

12.1.2 Passive, Desultory Protocol

Like the previous protocol, here, the user is expected to supervise the whole system's output to achieve a high-quality result. However, in this case the user can perform the amendments in a desultory order. This is especially useful when the elements of the output do not have a particular order or hierarchy between them.

Many different scenarios can fall under this category. However, in this book we analyze the case of information retrieval where the user types a natural language description of an object she is looking for. The system outputs a set of objects matching this description so the user can select which ones fit her needs and which do not. The system, then, implicitly rejects the objects with negative feedback and tries to fill the set with new objects taking into account the user preferences from the previous iterations. The procedure stops when the user chooses not to reject any further object from the set. The goal is to obtain such a set in the minimum number of interactions.

Relevant Image Search Engine (RISE) The user looks for images that can be described by a sentence in natural language. She selects the images she likes and asks for more images, until she decides that the retrieved results are good enough.

12.1.3 Active Protocol

Contrarily to passive protocols, active protocols are not oriented toward a perfect solution but a trade-off between effort and output quality. In this scenario, the system typically selects actively a part of the solution for which it has a low confidence, and asks the user for the correct solution. The procedure stops when a certain compromise between user effort and recognition rate has been achieved. Note that in this protocol the order of the output constituent is not specially relevant, since the whole output does not need to be supervised by the user. Instead, it is the system that decides the parts to be corrected and the order in which they are presented to the user.

GIMP-based Interactive Document Transcription (GIDOC) This system provides a user-friendly, integrated support for interactive-predictive layout analysis, line detection and handwriting transcription. GIDOC is designed to work with large collections of *homogeneous* documents (i.e. similar structure and writing styles). The system pursues a similar goal to MM-IHT. However, in this case the documents are transcribed by partially supervising system hypotheses. Furthermore, statistical models are constantly being updated with an increasing number of available annotated documents.

12.1.4 Prototype Evaluation

It is worth mentioning that the cost of a formal field study of this kind of systems is exceedingly high, since it typically involves expensive work by a panel of experts (for instance, qualified paleographers, professional translators, or trained computational linguists). For that reason, we decided to start doing a preliminary exploration, recruiting regular computer users instead. By now, the IHT prototype is the only demonstrator that has been formally evaluated, since it was the most advanced overall. However, we plan to do so with all prototypes in a near future. We encourage the reader to try the freely available demos and draw their own conclusions.

12.2 MM-IHT: Multimodal Interactive Handwritten Transcription

Transcribing handwritten text is a laborious task which, in general, is currently carried out manually. As the accuracy of automatic handwritten text recognizers improves, post-editing the output of these recognizers is foreseen as a possible alternative. However, given the high error rates of current approaches, post-editing is usually both inefficient and uncomfortable for the users. As alternative, an interactive-predictive paradigm has gained an increasing popularity due to promising empirical results that estimated considerable reductions of user effort.

In this section we introduce a web-based demonstrator of the interactive multimodal approach presented in Chap. 3. This new multimodal interactive approach for handwritten transcription, also known as MM-CATTI, MM-IHT, or simply IHT for short, is shown to work quite well by an implemented web-based demonstrator. The reader can access the online demo at http://cat.iti.upv.es/iht/.

12.2.1 Prototype Description

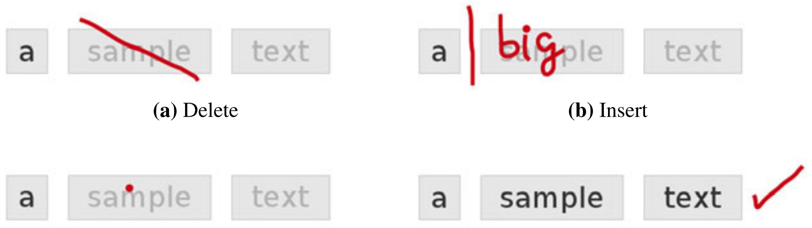
In this prototype [22] the server–engine communication is made through binary sockets [1]. A web application loads initially an index of all available pages in the document to be transcribed (Fig. 12.3). The user then navigates to a page and begins to transcribe the handwritten text images line by line (Fig. 12.5). She can make corrections with any kind of pointing device (e.g. touchscreen or stylus), and also she can use the keyboard. If pen strokes are available, the IHT engine uses an on-line HTR feedback subsystem to decode them. Finally, taking into account the decoded word and the off-line models, the engine responds with a suitable continuation to the prefix validated by the user.

On the other hand, keystrokes data directly interact with the aforementioned IHT engine. All corrections are stored in plain text logs on the server, so the user can retake them in any moment. Other client–server communications, such as managing logs or loading the sockets interface, are made via AJAX (Asynchronous JavaScript And XML), thus providing a richer interactive experience.

User Interaction Protocol

In the MM-IHT web-based demonstrator, the user is tightly involved with the transcription process, where following a preset protocol, she validates and/or corrects the HTR output during the process. Such a protocol, which rules this interaction process, is formulated in the following steps:

- The IHT engine proposes a full transcription of the input handwritten text line image.
- The user validates the longest prefix of the transcription which is error-free and enters some on-line touchscreen pen-strokes and/or some amendment keystrokes to correct the first error in the suffix.
- If pen strokes were available, an on-line HTR feedback subsystem is used to decode this input.
- In this way, a new extended consolidated prefix is produced based on the previous validated prefix, the on-line decoded word and the keystroke amendments. Using this new prefix, the IHT engine suggests a suitable continuation of it.
- These previous steps are iterated until a final, perfect transcription is produced.



(**a**) Delete (**b**) Insert

(**c**) Reject (**d**) Accept

Fig. 12.2 An example of pen strokes-related operations in the MM-IHT prototype

The interaction between the user and the system is not only limited to write the full correct word, but other different operations can be carried out using both pen-strokes and/or keystrokes. The types of operations that can be carried out are the following.

Substitute The first erroneous word is substituted by the correct word. The validated prefix consists of all the words preceding the substituted word and the new correct word.

Delete The incorrect word is deleted. The validated prefix consists of all the words preceding the deleted word plus the word that follows the deleted word.

Reject All the words that precede the incorrect word constitute the validated prefix. The system proposes a new suffix where the first word is different to the incorrect word.

Insert A new word is inserted. The validated prefix are all the word precedent the inserted word, the inserted word and the word that follows the inserted word.

Accept The proposed transcription is fully validated.

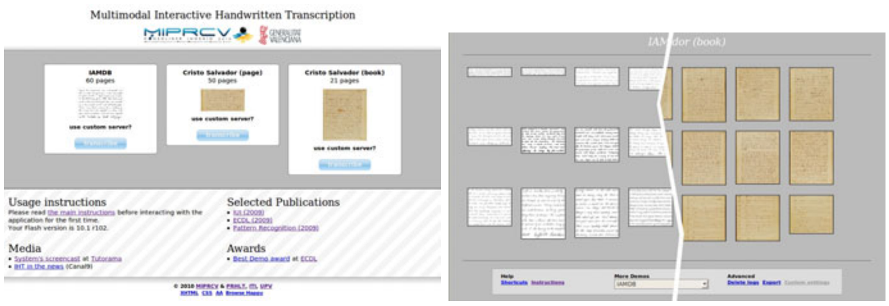
In Fig. 12.2 one can see different interaction modes making use of pen-strokes. The user can write down the correct word directly, make a diagonal line to delete an erroneous word, make a vertical line followed by a word for inserting that word, or make a single click to ask for another suitable suffix continuation [20].

12.2.2 Technology

IHT Engine

The IHT engine combines all the information received from the client and compute a suitable solution. It follows the approach presented in Chap. 3, where both online and offline HTR systems are based on HMMs and n-gram language models.

The offline system is implemented using word-graphs. These word-graphs are a pruned version of the Viterbi search trellis obtained when transcribing the whole image sentence. In order to make the system able to interact with the user in a time-efficient way, they are computed beforehand.



Multimodal Interactive Handwritten Transcription

IAMDB
60 pages
use custom server?

Cristo Salvador (page)
50 pages
use custom server?

Cristo Salvador (book)
21 pages
use custom server?

Usage instructions
Please read the main instructions before interacting with the application for the first time.
Your Flash version is 10.1 r102.

Media
- System's screencast at Tutorama
- IHT in the news (Canal9)

Selected Publications
- IUI (2009)
- ECDL (2009)
- Pattern Recognition (2009)

Awards
- Best Demo award at ECDL

© 2010 MIPRCV & PRHLT, ITI, UPV
XHTML CSS AA Browse Happy

(**a**) IHT Index (**b**) Page selection

Fig. 12.3 Once a corpus is selected on the IHT index (**a**), a thumbnail for each page of the document is presented to the user (**b**)

Once the user selects the line to transcribe, the client application send to the IHT engine the `set_source` message. The IHT engine loads the word-graph corresponding to the selected line and proposes a full transcription as explained in Sect. 2.2 of Chap. 2.

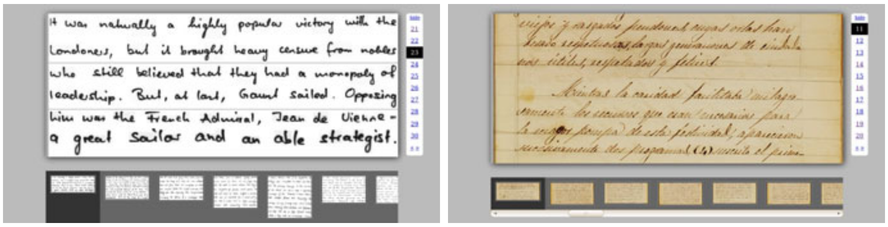
When the user makes some correction, if pen strokes are available, the IHT engine uses an on-line HTR feedback subsystem to decode them. After preprocessing and extracting the features, as is explained in Sect. 3.5.2, the pen strokes are decoded following the last scenario presented in Chap. 3, taking into account information derived from the validated prefix and the previous suffix as shown in Eq. (3.13).

Once the pen strokes have been decoded, a new prefix can be generated taking into account the validated prefix, the new decoded word and the operation that the user has carried out (substitution, deletion, insertion, etc.). Then, this new prefix is parsed in the off-line word-graph, and a suitable continuation is provided following techniques described in Chap. 3. It could happen that the prefix was not in the word-graphs, so, the error correcting parsing explained in Sect. 3.2.2 is applied.

Web Interface

The Web Interface is responsible for showing the user interface and capturing the user actions on the different modalities of interaction, i.e, keyboard and pen strokes. On the main page (Fig. 12.3a) of the demonstrator the user must choose one of the available documents to transcribe by clicking on the "*transcribe*" button. Also, by clicking on "*use custom server?*" link, the user can specify a custom IHT engine while her session is active.

Once the user has selected the document to transcribe, an index of all pages in the corpus appears, allowing the user to navigate to any page. In Fig. 12.3b we can see different pages from the IAMDB modern English corpus and from the Spanish 19th century handwritten document "Cristo Salvador".



It was naturally a highly popular victory with the Londoners, but it brought heavy censure from nobles who still believed that they had a monopoly of leadership. But, at last, Gaunt sailed. Opposing him was the French Admiral, Jean de Vienne - a great sailor and an able strategist.

(**a**) IAM database (**b**) Cristo Salvador corpus

Fig. 12.4 Corpora examples for the IHT prototype

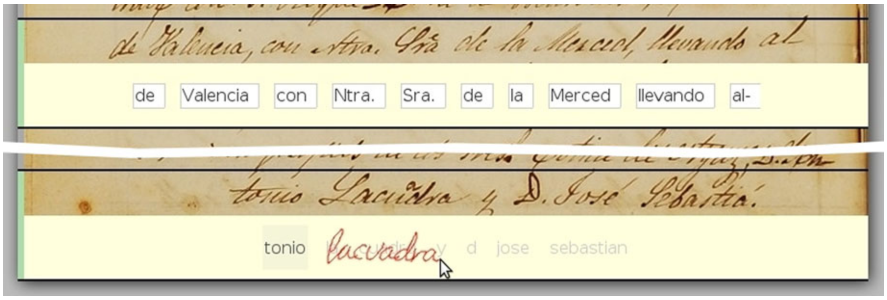


Fig. 12.5 Screenshot (detail) of two feedback modalities: keyboard input and e-pen

To begin with, once the user selects a thumbnail page from the index, the full page is loaded (Fig. 12.4). The center block is the page to transcribe itself. The tidy menu at the right side is a pagination item, to allow the user browsing all pages quickly. There is a page slider to allow a visual pagination of the selected corpus, and also a bottom menu intended to help the user with common tasks (such as closing session, changing the corpus, or displaying application shortcuts).

Then, the user can select a line from the current page by clicking on its image, and the system will propose an initial, full transcription. If no error is detected, the user chooses another text line image to transcribe. Otherwise, the user edits it interactively as explained before. All the corrections made by the user are stored in plain text logs on the web server. In this way, she can retake them at any moment.

12.2.3 Evaluation

The empirical tests on cursive handwritten tasks suggested that, using the IHT approach, considerable amounts of user effort could be saved with respect to both pure manual work and non-interactive, post-editing processing (see Sect. 3.1). While, of course, no definitive conclusions could be derived from these empirical tests, they clearly raised great expectations about the effectiveness and usability of this kind of interactive HTR technology. Therefore, in order to assess whether such expectations were in the right direction, we conducted a preliminary field study that compared the theoretical results with real users working with the implemented demonstrator.

Assessment Measures In interactive PR systems the importance of the well-known error rate metric is diminished per-se. Instead, the intention is to measure how well the user and the system work together. For this reason, to better judge the quality of the user transcriptions we used the two objective test-set-based measures introduced in Sect. 2.6: word error rate (WER) and word stroke ratio (WSR). Both metrics have proven to be useful for estimating the reduction in human effort that can be expected by using IHT with respect to using a conventional HTR system followed by human post-editing. On the other hand, in our subjective tests with real users we measured the time needed to fully transcribe each page with the different transcription possibilities (fully manual, post-editing, and IHT) as well as the residual WER (rWER)—defined as the WER which remains after the user has typed the transcriptions or corrected/accepted the transcriptions proposed by the system (this value is expected to be greater than zero due to human errors).

Corpus The corpus used on the experiments was the one identified as "Cristo Salvador" (CS), which has been previously presented in Sect. 3.6.1.

Participants Fourteen members from our Computer Science department volunteered to cooperate, aged 28 to 61 ($M = 37.3$, three females). Most of them were knowledgeable with handwriting transcription tasks, although none was a transcriber expert.

Apparatus The presented IHT demonstrator was modified to carry out the field study. We developed three kind of HTR engines to assist the document transcription: a trivial manual system, a post-editing system, and an interactive-predictive system. The user interface (UI) was common to all engines (see Fig. 12.4). Also, a logging mechanism was embedded in the web application. It allowed to register all user interactions in a fine-grained level of detail (e.g., keyboard and mouse events, client/server messages exchanging, etc.). The generated log files were reported in XML format for later postprocessing.

Procedure To carry out the experiments we used two pages (#45 and #46) from the test partition of the CS corpus. The WER and WSR of these pages are 24.5% and 23.0%, respectively. Participants accessed the web-based application via a special URLs that were sent to them by email. In order to familiarize with the UI, users informally tested each transcription engine with some test pages, different from the ones reserved for the user-test. Only three participants were aware of the existence of such a demonstrator prior to the study. Then, people transcribed the two user-test pages; each one with the three transcription engines. The two user-test pages for the field study were selected because they had very similar test-set-based performance

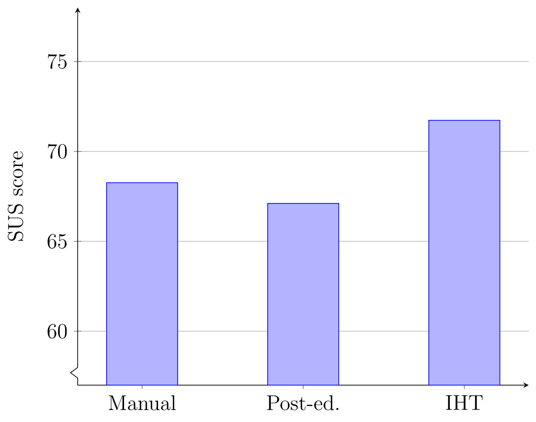


Fig. 12.6 User satisfaction, according to the SUS questionnaire

metrics (CS page #45: WER = 24.35%, WSR = 23.07%; CS page #46: WER = 24.69%, WSR = 23.04%; respectively). It is important to remark that nobody saw these pages before the study. For that reason, it is clear that the engine that were tested at first would lead to poorer results than the rest of engines—in the next trials users would need less effort in reading the image lines. Thus, to avoid possible biases due to human learnability, page #45 was transcribed in this order: manual, post-edition, and IHT. Then the order was inverted for page #46. Finally, participants filled out a SUS questionnaire for each engine, including a text field to enter free comments and ideas about their testing experience as well as give some insights about possible enhancements and/or related applicability.

Design A within-subjects repeated measures design was carried out. We tested two conditions: transcribing a page when it has not been seen before and when it has been seen at least once. Since both normality assumptions and homocedasticity in the data were met, we performed a four-way ANOVA experiment. The independent variables were time, rWER, and WSR, respectively.

Discussion of Results

In sum, we can assert that regarding effectiveness there are no significative differences, i.e., users can achieve their goals with any of the proposed systems. However, in terms of efficiency the IHT system seems to be the better choice. Regarding user satisfaction, Fig. 12.6 clearly shows that IHT is the most preferable of the three options. Now let us delve into a more detailed analysis in order to shed more light to the obtained results.

For the sake of saving time for the users, the post-editing engine was tested always as the second option. However, we must emphasize that the daily use of any system designed to assist handwriting transcription would involve not having seen previously any of the pages (users usually read a page once and at the same time they just transcribe it).

- *Analysis of Time*: Overall, post-editing attained the best time. This was expected, since for both user-test pages such system was tested after users had read each page once. However, the manual and IHT engines were tested under the same conditions, and we observed that IHT is approximately 1 minute faster. In any case, these differences are not statistically significant ($F_{2,76} = 1.98$, $p = 0.14$). In general, the last used system achieves the best time, because the user already knows the text. The remarkable result is that when the user reads a page in first place the chosen engine is not determinant, because one must spend time to accustom to the writing style, interpreting the calligraphy, etc. Additionally, though, we might count the number of times that one system outperforms the other for a given user, and thereby, measure the *probability of improvement* (POI) [4]. In this case the POI of the IHT engine with respect to the manual engine is 53%, and 42% regarding to post-edition.
- *Analysis of rWER*: Overall, IHT was the best choice regarding to residual WER in all situations, although the differences are not statistically significant ($F_{2,75} = 0.67$, $p = 0.51$). The interesting observation is that IHT is the most stable of the systems, even better than when using the manual engine on an already read page (the post-editing system was expected to perform similarly, since it was tested in both conditions in second place). We must recall that the more stable in rWER a system is, the fewer residual transcription errors are expected. In this case, considering the first time that the user reads a page, the POI of the IHT engine over the manual engine is 69%, and 68% with respect to post-edition.
- *Analysis of WSR*: Interestingly, the WSR when using the manual engine was below 100%, since there are inherent errors (some users were unable to correctly read all the lines). That means that some users wrote less words in their final transcriptions than they really should have to. Overall, ANOVA showed that IHT was the best performer in both conditions. Differences were statistically significant ($F_{2,76} = 1014.71$, $p < 0.0001$, $\eta^2 = 26.7$). This means that the number of words a user must write and/or correct under the IHT paradigm is always lower than with any of the other systems. Additionally, this fact increases the probability of achieving a high-quality final transcription, since users perform fewer interactions and are prone thus to less errors. In this case the POI of the IHT engine regarding the manual engine is 100%, and 65% with respect to post-edition. It is also interesting to note that, on average, the real WSR achieved by the participants is fairly close to the objective user-test based estimates for the same pages and even closer to the theoretical test-set estimates overall.
- *Analysis of User Subjectivity*: Regarding user responses to the SUS questionnaire, while no statistically significant differences were achieved ($F_{2,32} = 0.11$, $p = 0.9$), there is a clear tendency in favor of IHT (see Fig. 12.6). What is more, participants chose the manual engine over post-editing most of the time. This would explain why, considering difficult transcription tasks, users generally refuse to post-edit the output of a conventional HTR system, since it has too many recognition errors.

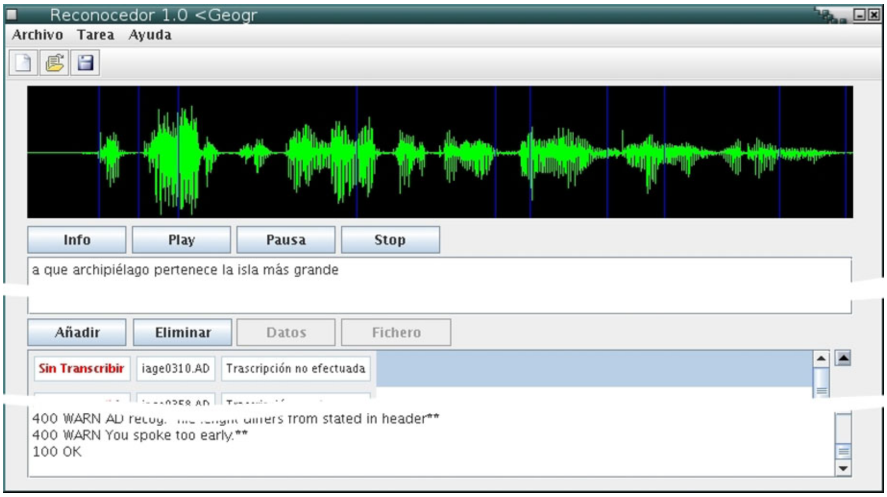
Most of the users' comments were alas related to the web UI rather than the transcription engines themselves. Some included "*when clicking on a text field, the whole word is selected*", "*it is hard to remember some [keyboard] shortcuts*", or "*a clear and visual user manual would allow not having to learn almost anything before using the system.*" Additionally, four users complained about the segmentation of lines, which "*made especially difficult reading those images where words had many ascenders/descenders.*" On the other hand, three users noticed that punctuation chars did not contribute to improve predictions in the IHT system. In fact, they were removed from the language models when training the IHT engine, since we used bi-grams and punctuation chars do not improve notably the predictions.

Limitations of the Study and Conclusion

There are a number of reasons why we were unfortunately not able to achieve statistically significant differences between the three tested engines in some cases. First, most of the participants had never faced neither any of the implemented engines nor the web UI before the study, so it is expected a logical learning curve prior to using such systems in a daily basis. Second, the web interface was just a prototype, and it is well known that a careful design of the UI is a primary factor to tap the possibilities of the IHT technology. Third, the pages were really deteriorated, making thus the reading difficult for the users. For that reason, there is a great difference between the first time that a user had to transcribe a page and the subsequent attempts. Fourth, the post-edition engine was not tested under the same circumstances as the other two engines, i.e., it was always used in second place. Thus an important bias was introduced in the comparison of the systems—although it was not the case for manual and IHT engines, which were both tested in the same conditions. Finally, due to the limited size of user sample, we had to include the outliers in the evaluation. However, despite the above-mentioned limitations, there is a comprehensible tendency to choose the IHT paradigm over the other two systems. Additionally, as observed, the probability of improvement of an IHT engine over manual transcription or post-edition makes this paradigm worthwhile.

12.3 IST: Interactive Speech Transcription

The transformation audio speech signals into text is an important activity in many official institutions (e.g. the European Union parliament, U.N. sessions, Catalan and Basque parliaments in Spain, etc.) and private companies. However, speech recognition systems are not perfect. Special cases always appear which are not accounted for by automatic systems, leading to inaccurate and/or inadequate transcriptions. In practice, such systems generally need human postprocessing to correct the possible errors incurred by the system.



a que archipiélago pertenece la isla más grande

```
400 WARN AD recog. ... differs from stated in header**
400 WARN You spoke too early.**
100 OK
```

Fig. 12.7 MM-IST User Interface

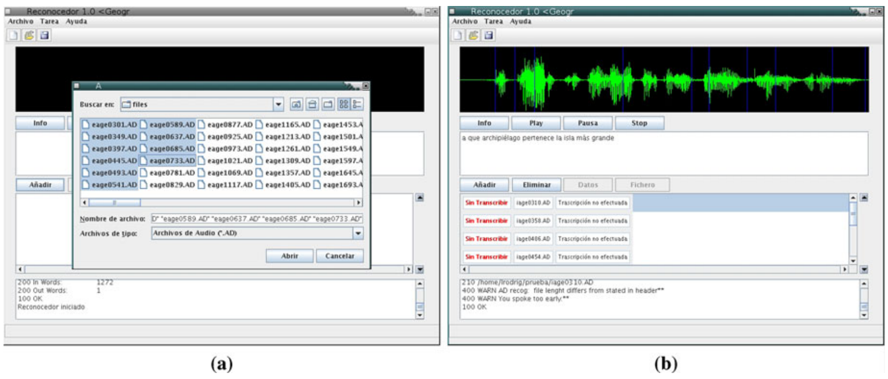
In this section we introduce an Interactive Speech Transcription (IST) prototype [19] developped within the general Interactive Pattern Recognition approach presented in Chap. 4. This prototype provides an interactive framework for obtaining high-quality results while increasing the productivity with respect to the classical postprocessing techniques of speech transcription.

12.3.1 *Prototype Description*

A Java interface provides all the interaction mechanism to allow an efficient communication between the user and the prototype (see Fig. 12.7). Initially, the interface ask the user to create a new transcription project or to use a previously created one. A project is a set of audio files to transcribe. Along with these files a project stores the transcriptions already obtained. The main interface screen shows, on the one hand, the list of files in the current project so that the user can choose the file to be transcribed (see Fig. 12.8). On the other hand, the waveform of the current file is shown in a graphic panel and a text field is used to show the current system suggestion along with the different user interactions performed (see Fig. 12.9). The user can interact with this text field by using either the mouse or the keyboard to select the error-free prefix. In addition, the keyboard is used to amend the system suggestions.

User Interaction Protocol

It is similar to the protocol used in IHT discussed in the previous section.



eage0301.AD eage0589.AD eage0877.AD eage1165.AD eage1453.A
eage0349.AD eage0637.AD eage0925.AD eage1213.AD eage1501.A
eage0397.AD eage0685.AD eage0973.AD eage1261.AD eage1549.A
eage0445.AD eage0733.AD eage1021.AD eage1309.AD eage1597.A
eage0493.AD eage0781.AD eage1069.AD eage1357.AD eage1645.A
eage0541.AD eage0829.AD eage1117.AD eage1405.AD eage1693.A

Nombre de archivo: D" "eage0589.AD" "eage0637.AD" "eage0685.AD" "eage0733.AD"
Archivos de tipo: Archivos de Audio (*.AD)

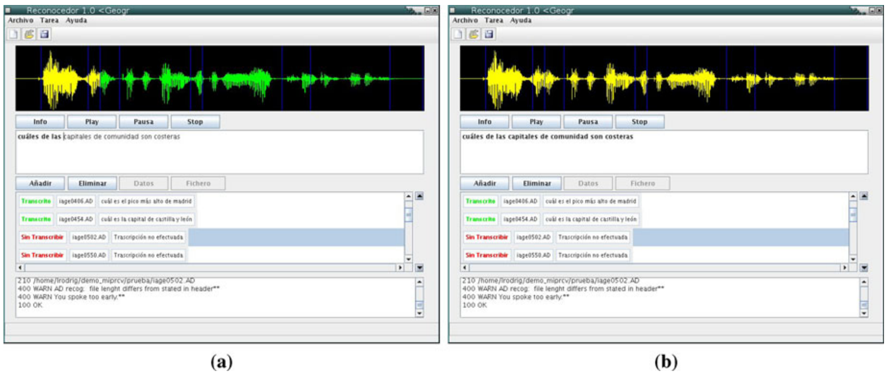
```
200 In Words:      1272
200 Out Words:     1
100 OK
Reconocedor iniciado
```

a que archipiélago pertenece la isla más grande

Sin Transcribir	iage0310.AD	Trascripción no efectuada
Sin Transcribir	iage0358.AD	Trascripción no efectuada
Sin Transcribir	iage0406.AD	Trascripción no efectuada
Sin Transcribir	iage0454.AD	Trascripción no efectuada

```
210 /home/lrodrig/prueba/iage0310.AD
400 WARN AD recog:  file lenght differs from stated in header**
400 WARN You spoke too early.**
100 OK
```

Fig. 12.8 Loading a set of speech utterances to transcribe



(a) cuáles de las capitales de comunidad son costeras

(b) cuáles de las capitales de comunidad son costeras

Transcrito	iage0406.AD	cuál es el pico más alto de madrid
Transcrito	iage0454.AD	cuál es la capital de castilla y león
Sin Transcribir	iage0502.AD	Trascripción no efectuada
Sin Transcribir	iage0550.AD	Trascripción no efectuada

```
210 /home/lrodrig/demo_miprcv/prueba/iage0502.AD
400 WARN AD recog:  file lenght differs from stated in header**
400 WARN You spoke too early.**
100 OK
```

Fig. 12.9 Using the IST prototype, discerning those utterance parts that are already transcribed (*yellow wave*) from those that are not (*green wave*)

12.3.2 Technology

Prediction Engine

The prediction engine is a general-purpose, speaker-independent, continuous speech recognizer (ATROS). This engine is a standalone application capable of recognizing speech in real time from a microphone input or from an audio file.

Given an input utterance, the recognizer initially preprocess the signal by performing a border detection algorithm and a pre-emphasis filter to increase the magnitude of the high frequencies present in the speech signal. Next, a feature extraction is performed to obtain a MFCC (Mel Frequency Cepstral Coefficients) vector for each frame in the input signal. Then, the energy along with the first and second derivatives are added to build the final feature vector for the frame. Once the feature extraction has been preformed, the recognition process is carried out.

The recognizer uses three-state, left-to-right Hidden Markov Models as acoustic models. Finite-state automata are used as lexical entries and, finally, language models are implemented as finite-state networks. The Viterbi algorithm [8] is used to find the most probable path on the integrated network (acoustic+lexical+language models). Finally, the word sequence associated with this optimal path is returned as the recognition result.

Communication Module

Since the prediction engine and the user interface have been developed in different languages, a communication module is needed to send the user feedback from the user interface to the engine and to collect the different predictions. This module is currently implemented by adding a communication API to the recognizer. This API is implemented by using the Java Native Interface (JNI) that allows a Java program to call C/C++ functions. In this way, this API is included into the recognizer's building process so that the interface can make the proper calls during runtime. Three main functions are defined in the API which are responsible for initializing the recognizer, setting a new user prefix and requesting a new prediction from the engine. It is worth mentioning that currently the CAT-API [1] is being implemented in order to deploy a web-based version of this prototype (see reference functions in Sect. 12.1.1).

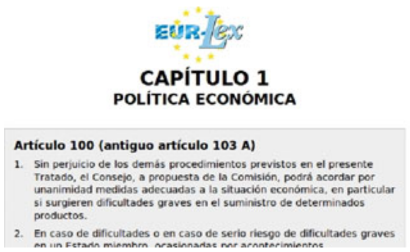
12.3.3 Evaluation

To assess the IST approach discussed above, different experiments were carried out. The reader is redirected to Sects. 4.6 and 4.6.3. The empirical results showed a clear effectiveness of the IST approach in the tested corpora.

12.4 IMT: Interactive Machine Translation

The WWW with its universal access to information and instant communication between users has created a physical and geographical freedom for translators that was inconceivable in the past [6]. Computer-Assisted Translation (CAT) is an alternative approach to Machine Translation, integrating human expertise into the automatic translation process. Interactive Machine Translation (IMT) [2] can be considered as a special type of the CAT paradigm. The IMT framework was later extended in [24], where the possibility of rejecting a given suffix was introduced. The above-mentioned IMT framework is shown to work quite well by a web-based architecture.

In this section we bring to practice the theory discussed in Chap. 6. In similar web-based natural language processing systems [21, 23], the user's feedback has shown to improve system accuracy, and increase both system ergonomics and

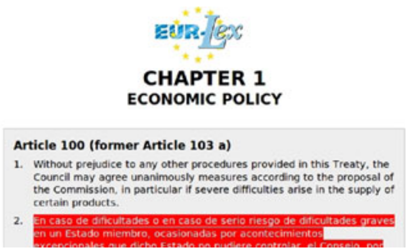


CAPÍTULO 1
POLÍTICA ECONÓMICA

Artículo 100 (antiguo artículo 103 A)

1. Sin perjuicio de los demás procedimientos previstos en el presente Tratado, el Consejo, a propuesta de la Comisión, podrá acordar por unanimidad medidas adecuadas a la situación económica, en particular si surgieren dificultades graves en el suministro de determinados productos.
2. En caso de dificultades o en caso de serio riesgo de dificultades graves en un Estado miembro, ocasionadas por acontecimientos

(**a**) Source document example, created from the EuroParl corpus.



CHAPTER 1
ECONOMIC POLICY

Article 100 (former Article 103 a)

1. Without prejudice to any other procedures provided in this Treaty, the Council may agree unanimously measures according to the proposal of the Commission, in particular if severe difficulties arise in the supply of certain products.
2. En caso de dificultades o en caso de serio riesgo de dificultades graves en un Estado miembro, ocasionadas por acontecimientos excepcionales que dicho Estado no pudiere controlar, el Consejo, por

(**b**) Translated example document, preserving original format and highlighting non-translated sentences.

Fig. 12.10 Translating documents with an IMT system

user's acceptability. Since the users operate within a web browser, this prototype also provides crossplatform compatibility and requires neither computational power nor disk space on the client machine. The reader can access the online demo at http://cat.iti.upv.es/imt/.

12.4.1 Prototype Description

This prototype is not intended to be a production-ready application. Rather, it provides an intuitive interface which aims at showing the functionality of an IMT system. Thus, two main aspects on which the architecture has been built are accessibility and flexibility. The former is necessary to reach a larger number of potential users. The latter allows researchers to test different techniques and interaction protocols reducing the implementation effort.

The proposed system [14] coordinates client-side scripting with server-side technologies, by using the CAT-API library [1]. At first, the web interface loads an index of all available corpora. Each corpus consists of a web document that is parsed from an automatically generated Translation Memory eXchange (TMX) file.[1] In this way, it is possible to recover the original format from source document and apply it back to the translated version of that document (see Fig. 12.10).

The user then navigates to a page and begins to translate the document by text segments. She can make corrections with the keyboard and also accomplish some mouse operations. User's feedback is then processed by the IMT engine. The IMT User Interface appears in Figs. 12.12 and 12.13.

User Interaction Protocol

The protocol that rules the interaction process has the following steps:

1. The system proposes a full translation of the selected text segment.

[1]TMX is an open XML standard for the exchange of translation documents.

source	Para ver la lista de recursos:
interaction-0	To view the resources list:
interaction-1	To view |a list of resources
interaction-2	To view a list$\boxed{\text{i}}$ng resources:
interaction-3	To view a listing $\boxed{\text{o}}$f resources:
acceptance	To view a listing of resources:

Fig. 12.11 IMT session to translate a Spanish sentence into English. On interaction-0, the system suggests a translation. On interaction-1, the user moves the mouse to validate (VP) the first eight characters "To view" and rejects (R) the next word; then the system suggests completing the sentence with "a list of resources". On interaction-2 the user moves again the mouse to accept the next six characters and presses the key *i* (KS). interaction-3 is similar to interaction-2. Finally, the user completely accepts (A) the present suggestion

2. The user validates the longest prefix of the translation which is error-free and/or corrects the first error in the suffix. Corrections are entered by amendment keystrokes or mouse-click operations.
3. In this way, a new extended consolidated prefix is produced based on the previous validated prefix and the interaction amendments. Using this new prefix, the system suggests a suitable continuation of it.
4. Steps 2 and 3 are iterated until the user-desired translation is produced.

System Interaction Modes

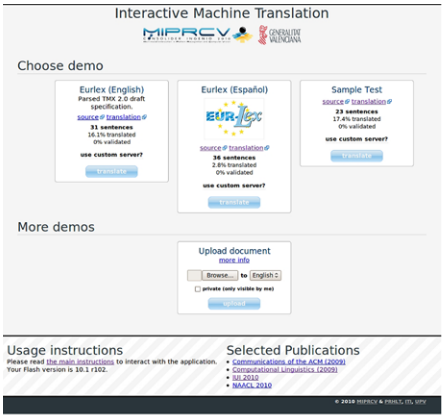
Our proposed system works both at full-word and character level, that is, the user can introduce modifications to the system by interacting with minimum and atomic text parts, respectively. The types of operations that can be carried out are grouped in four categories:

Validate prefix (VP) The text at the left of the mouse pointer is validated.
Key/Mouse stroke (KS/MS) The next character of the desired translation is inserted with the keyboard/mouse.
Reject (R) The text to the right of the pointer is wrong.
Accept (A) The text is finally correct.

Figure 12.11 shows an example of a typical IMT session involving the four interaction modes that have been described above (key stroke operations are represented by framed boxes and reject operations are represented by vertical bars).

12.4.2 Technology

In this section we describe the technology used to implement the IMT server and the web interface needed to communicate both client and server sides. On the one



Interactive Machine Translation

Choose demo

Eurlex (English)
Parsed TMX 2.0 draft specification.
source translation
31 sentences
16.1% translated
0% validated
use custom server?
translate

Eurlex (Español)
source translation
36 sentences
2.8% translated
0% validated
use custom server?
translate

Sample Test
source translation
23 sentences
17.4% translated
0% validated
use custom server?
translate

More demos

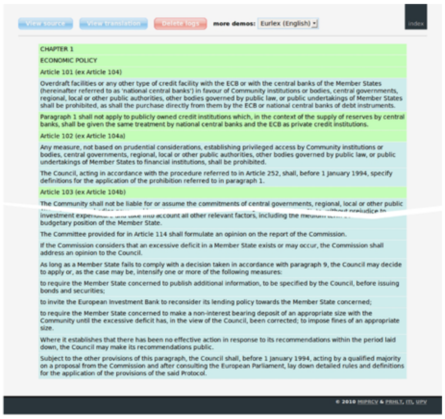
Upload document
more info
Browse... to English
private (only visible by me)
upload

Usage instructions
Please read the main instructions to interact with the application.
Your Flash version is 10.1 r102.

Selected Publications
- Communications of the ACM (2009)
- Computational Linguistics (2009)
- IUI 2010
- NAACL 2010

© 2010 MIPRCV & PRHLT, ITI, UPV

(a) IMT Index



View source | View translation | Delete logs | more demos: Eurlex (English) | index

CHAPTER 1

ECONOMIC POLICY

Article 101 (ex Article 104)

Overdraft facilities or any other type of credit facility with the ECB or with the central banks of the Member States (hereinafter referred to as 'national central banks') in favour of Community institutions or bodies, central governments, regional, local or other public authorities, other bodies governed by public law, or public undertakings of Member States shall be prohibited, as shall the purchase directly from them by the ECB or national central banks of debt instruments.

Paragraph 1 shall not apply to publicly owned credit institutions which, in the context of the supply of reserves by central banks, shall be given the same treatment by national central banks and the ECB as private credit institutions.

Article 102 (ex Article 104a)

Any measure, not based on prudential considerations, establishing privileged access by Community institutions or bodies, central governments, regional, local or other public authorities, other bodies governed by public law, or public undertakings of Member States to financial institutions, shall be prohibited.

The Council, acting in accordance with the procedure referred to in Article 252, shall, before 1 January 1994, specify definitions for the application of the prohibition referred to in paragraph 1.

Article 103 (ex Article 104b)

The Community shall not be liable for or assume the commitments of central governments, regional, local or other public [illegible] without prejudice to investment expen[illegible] into account all other relevant factors, including the medium [illegible] budgetary position of the Member State.

The Committee provided for in Article 114 shall formulate an opinion on the report of the Commission.

If the Commission considers that an excessive deficit in a Member State exists or may occur, the Commission shall address an opinion to the Council.

As long as a Member State fails to comply with a decision taken in accordance with paragraph 9, the Council may decide to apply or, as the case may be, intensify one or more of the following measures:

to require the Member State concerned to publish additional information, to be specified by the Council, before issuing bonds and securities;

to invite the European Investment Bank to reconsider its lending policy towards the Member State concerned;

to require the Member State concerned to make a non-interest bearing deposit of an appropriate size with the Community until the excessive deficit has, in the view of the Council, been corrected; to impose fines of an appropriate size.

Where it establishes that there has been no effective action in response to its recommendations within the period laid down, the Council may make its recommendations public.

Subject to the other provisions of this paragraph, the Council shall, before 1 January 1994, acting by a qualified majority on a proposal from the Commission and after consulting the European Parliament, lay down detailed rules and definitions for the application of the provisions of the said Protocol.

© 2010 MIPRCV & PRHLT, ITI, UPV

(b) Demo Layout

Fig. 12.12 IMT User Interface

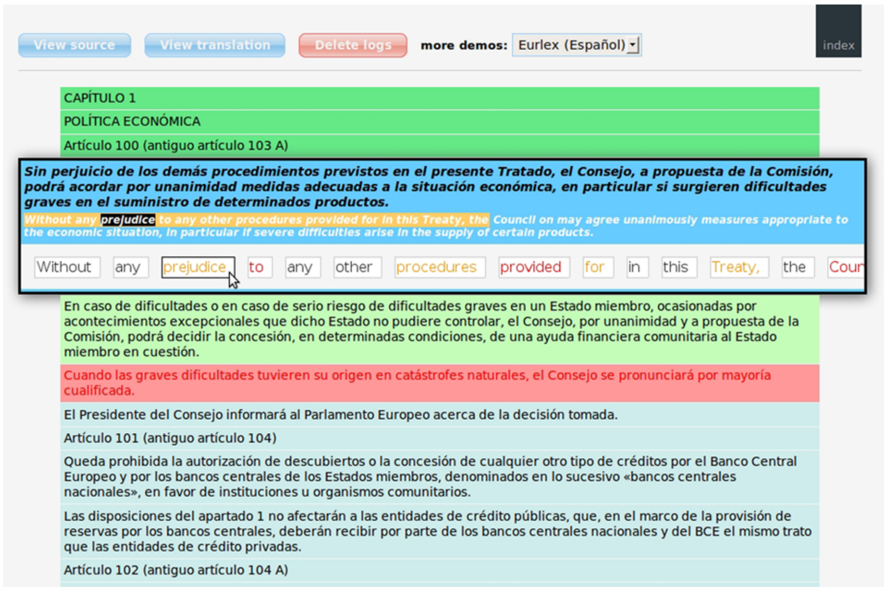
hand, the client provides a user interface which uses the above-mentioned CAT-API library to communicate with the IMT engine through the Web. The hardware requirements in the client are very low, as the translation process is carried out remotely on the server. So virtually any computer (including netbooks, tablets or 3G mobile phones) should be enough to use this system. On the other hand, the server, which is unaware of the implementation details of the IMT client, uses and adapts the statistical models that are used to perform the translation.

Interactive CAT Server

The IMT server is based on the statistical phrase-based translation approach [9] to generate the suffixes completing the consolidated prefixes given by the user. State-of-the-art SMT systems use a log-linear combination of features to generate their translations. Among the features typically used in this log-linear combination, the most important of them are the statistical language model, which is implemented by means of n-gram language models, and the statistical translation model, which is implemented by means of phrase-based models. Standard SMT systems require little modification for its use in the IMT framework [2].

Web Interface

Client-Server communication is made via asynchronous HTTP requests, providing thus a richer interactive experience, and Server-Engine communication is made through binary sockets. All corrections are stored in plain text logs on the server, so the user can retake them at any moment, also allowing other users to help to translate the full documents.



CAPÍTULO 1

POLÍTICA ECONÓMICA

Artículo 100 (antiguo artículo 103 A)

Sin perjuicio de los demás procedimientos previstos en el presente Tratado, el Consejo, a propuesta de la Comisión, podrá acordar por unanimidad medidas adecuadas a la situación económica, en particular si surgieren dificultades graves en el suministro de determinados productos.

Without any prejudice to any other procedures provided for in this Treaty, the Council on may agree unanimously measures appropriate to the economic situation, in particular if severe difficulties arise in the supply of certain products.

Without any prejudice to any other procedures provided for in this Treaty, the Coun

En caso de dificultades o en caso de serio riesgo de dificultades graves en un Estado miembro, ocasionadas por acontecimientos excepcionales que dicho Estado no pudiere controlar, el Consejo, por unanimidad y a propuesta de la Comisión, podrá decidir la concesión, en determinadas condiciones, de una ayuda financiera comunitaria al Estado miembro en cuestión.

Cuando las graves dificultades tuvieren su origen en catástrofes naturales, el Consejo se pronunciará por mayoría cualificada.

El Presidente del Consejo informará al Parlamento Europeo acerca de la decisión tomada.

Artículo 101 (antiguo artículo 104)

Queda prohibida la autorización de descubiertos o la concesión de cualquier otro tipo de créditos por el Banco Central Europeo y por los bancos centrales de los Estados miembros, denominados en lo sucesivo «bancos centrales nacionales», en favor de instituciones u organismos comunitarios.

Las disposiciones del apartado 1 no afectarán a las entidades de crédito públicas, que, en el marco de la provisión de reservas por los bancos centrales, deberán recibir por parte de los bancos centrales nacionales y del BCE el mismo trato que las entidades de crédito privadas.

Artículo 102 (antiguo artículo 104 A)

Fig. 12.13 IMT User Interface, using confidence measures to mark those words that presumably need to be verified or corrected by the user

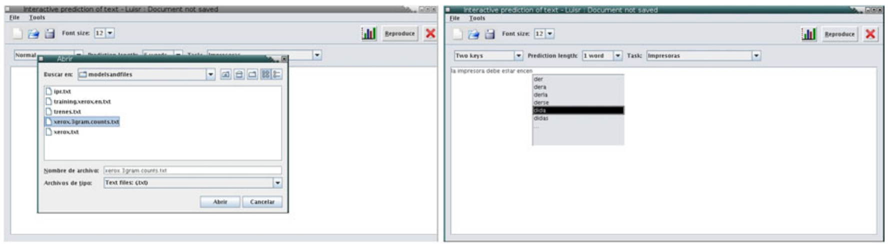
12.4.3 Evaluation

Experimental results were carried out on different tasks (see Sect. 6.4). It is worth noting that the reported experiments simulate real users to measure the effort required to generate error-free translations. The results suggested that the proposed system can substantially reduce the typing effort needed to produce a high-quality translation of a given source text with respect to the effort needed to simply type the whole translation. Specifically, 80% and 70% typing effort reductions were obtained for the Xerox and the European Union corpora, respectively.

12.5 ITG: Interactive Text Generation

The time spent in typing (including, as well, thinking about what is going to be typed) is quite high in many real-world situations. In addition, in some situations, typing becomes a slow and uncomfortable task. For example, in some devices, such as mobile phones, no suitable input mechanisms are available. Moreover, some disabled people are not able to achieve a fast enough typing speed and, unfortunately, in some cases, this is the only way for them to communicate.

In this section we present a system that takes all these considerations into account for assisting users in text generation tasks. Previous approaches to this topic can be found in the literature. Most of them just attempt to predict the next single word



Interactive prediction of text - Luisr : Document not saved

Font size: 12

Normal

Abrir

Buscar en: modelsandfiles

Nombre de archivo: xerox.3gram.counts.txt

Archivos de tipo: Text files: (.txt)

Two keys

Prediction length: 1 word

Task: Impresoras

la impresora debe estar encen

(a) (b)

Fig. 12.14 Loading a task-specialized corpus (**a**) and interacting with the ITG prototype (**b**)

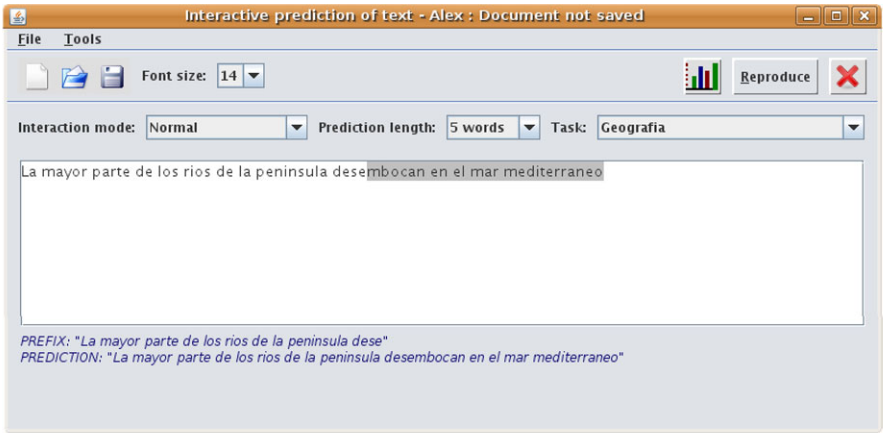
and/or to complete the characters of each new word being typed by the user [30]. Other systems just focus on measuring the accuracy of off-line text predictions [3]. In the system presented in this section, however, a more general setting is considered where not only single words, but multi-word fragments or even full sentences, are predicted interactively.

12.5.1 Prototype Description

A prototype for interactive text prediction, based on the IPR framework, has been implemented. The main prototype features are summarized in the following list.

- A user profile is maintained to store preferences and interaction statistics.
- A language or a domain can be selected (that is, the language model to be used).
- Different interaction modes (see Sect. 12.5.1) and prediction options (word or character level, prediction length, etc.) are available.
- Predictions are integrated into the main text editor.
- The system maintains statistics about the usage.
- An additional tool can be used to record a user session, keeping track of all the user and system actions. This is aimed at using the prototype in play-back mode in order to analyze the behavior of both the user and the system.

During its normal operation mode, the system is also able to take advantage of the generated text to improve the statistical prediction models. The rationale behind this idea is to adapt the system to a specific user so that the system is able to learn the user's writing style (or a new task or domain, for instance). Moreover, the system could be still used if no previously trained models were available. This way, all the learning process will perform on-line. Initially, the system predictions will be useless but, as the user generates more and more text, these predictions will be improved as a consequence of this on-line training process. Eventually, we will have a prediction model specifically trained for the task being currently solved.



La mayor parte de los rios de la peninsula desembocan en el mar mediterraneo

PREFIX: "La mayor parte de los rios de la peninsula dese"
PREDICTION: "La mayor parte de los rios de la peninsula desembocan en el mar mediterraneo"

Fig. 12.15 Example of a sentence being edited using the keyboard-based interaction mode. The text in *white* background is the prefix consolidated so far, and the text in *grey* background is the current system suffix suggestion

System Architecture

The ITG prototype has been divided into two different subsystems: the User Interface (UI, see Fig. 12.14) and the prediction engine. The UI allows the user to interact with the system, as well as set the different configuration options. The main component in this UI is a text editor where the system displays the proposed suffixes while the user is typing. She can accept, modify or discard these suffix-suggestions just by using the keyboard or the mouse. This interface also displays statistics on the system usage and can replay a previously saved session. The prediction engine, on the other hand, searches for the most suitable suffix (or set of suffixes) according to the current prefix.

This division allows the prediction engine to be deployed in a separate computer, or even implement several engines on a distributed computing system (e.g., each computer can host a single language model). Taking this fact into account, the prediction engine can be placed in a server (or servers) allowing the use of thin clients (PDAs, mobile phones, etc.) to have access to this technology. The system is based on the MIPR CAT client–server architecture (see Fig. 12.1) using socket interfaces and the TCP protocol for communication among the different subsystems.

User Interaction Protocol

Given the text typed so far, the system produces a prediction that the user reads until a mistake is found. Next, such a mistake is corrected and the system provides a new continuation taking this correction into account (as can be seen in Fig. 12.15). In this scenario, we are interested in minimizing the number of user interactions (i.e. to save user effort). Under this criterion, a simple greedy strategy consisting in suggesting the most probable word given the text typed (prefix) so far turns out to be the optimal strategy for this scenario [13]. It is worth noticing that this can be easily extended to multi-word predictions by considering each predicted word as part of the current prefix, and repeating this process until reaching the desired amount of predicted words.

System Interaction Modes

Two different interaction modes have been implemented. One may note that, in addition to these modes, as it was previously discussed, the system can work at both word or character level.

The first option (Fig. 12.15) considers that a keyboard (or a similar input device) is being used. The user can achieve high typing performance and the predictions should not slow down the process. This mode predicts just one suffix, which is directly inserted after the text typed so far. This suffix is highlighted to indicate that it is not still validated. The user can validate part of the suffix, the whole suffix or ignore it completely. In any case, the system suggestions do not interrupt the user, who can keep typing all the time. It is not clear whether this modality is really useful for a normal typing session, although it can be really helpful for slow-typing users or to type text in a non-native language, for instance. In these cases, the system could be seen as a language generation assistant rather than a typing effort saver.

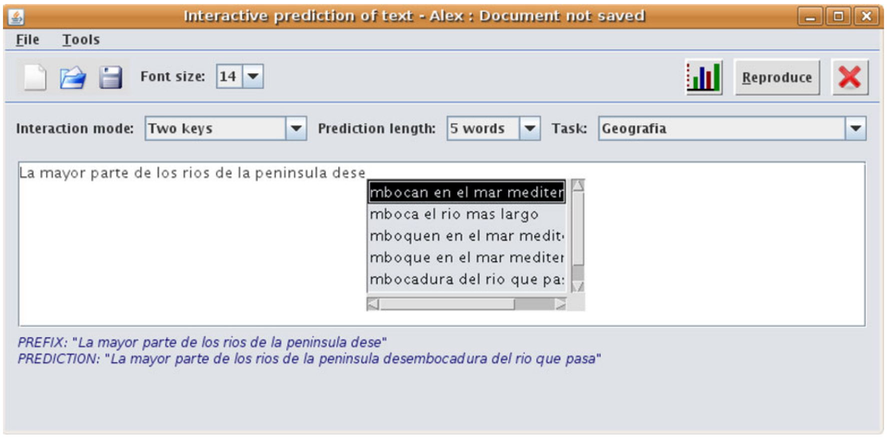
The second option (Fig. 12.16) allows the system to be used with just two keys. This option is planned for users who have some kind of physical impairment or with highly constrained input interfaces. In this case, a set of alternative suffixes (in the form of n-best list) is shown when pressing one of the two keys. At this point, the user can navigate (using the same key) within this list and choose any suggestion by pressing the other key. If no suitable continuations are found in the list, a new listing containing all the characters in the alphabet is displayed. In this way, the user can select an alternative continuation and, as soon as the user picks one of these characters, the system reacts by predicting a new suffix.

These two interaction modes have been implemented considering that the system has a standard keyboard. However, they can be easily adapted to any kind of hardware that allows a multi-state input (first option) or a two-state input (second option). For instance, the two-key option can be adapted with little effort to be used with a two-button mouse (or any kind of two-positions device).

12.5.2 Technology

The prototype is composed of three different subsystems:

- An automatic text predictor. Based on n-gram language models and implemented in C++.


Interactive prediction of text - Alex : Document not saved
File
Tools
Font size: 14
Reproduce
Interaction mode: Two keys
Prediction length: 5 words
Task: Geografia

La mayor parte de los rios de la peninsula dese

mbocan en el mar mediter
mboca el rio mas largo
mboquen en el mar medit
mboque en el mar mediter
mbocadura del rio que pa:

PREFIX: "La mayor parte de los rios de la peninsula dese"
PREDICTION: "La mayor parte de los rios de la peninsula desembocadura del rio que pasa"

Fig. 12.16 Example of sentence being edited using the two-key-based interaction mode. Several suffixes are shown in a contextual menu at the current cursor position

- A Graphical User Interface. This interface provides all the interaction engine to allow an efficient communication between the user and the prototype. The interface has been implemented in Java.
- A communication module between the text predictor and the graphical interface, that sends the information derived from the user interaction to the predictor. This module is implemented by following a server-client approach and is based on the use of network sockets.

The technical details of the language modeling and searching strategy were detailed in Sects. 10.2.1 and 10.2.2, respectively.

12.5.3 Evaluation

In order to assess whether if the prototype could facilitate the user interaction with the computer, we performed some preliminary (and informal) experiments involving real users. Test users were split into groups according to their general experience with computer applications and to their skills about using standard input interfaces. Several sessions were performed by each user and finally a questionnaire was answered. Preliminary evaluations have shown that the prototype is highly usable in most of the cases. Nevertheless, those users with little experience with computers usually needed some kind of assistance.

Several experiments have been performed considering different tasks and calculating their corresponding metrics. The main evaluation was performed offline by using test sets and computing the Key and Word Stroke Ratio (KSR and WSR) metrics (see Sect. 2.6). In all tested scenarios the estimated effort saving was significant. For instance, in the simple EuTrans task, the system achieved a KSR of 14% and a WSR of 50%, and for printers' technical manuals (Xerox corpus) the KSR and WSR were 19% and 66%, respectively.

12.6 MM-IP: Multimodal Interactive Parsing

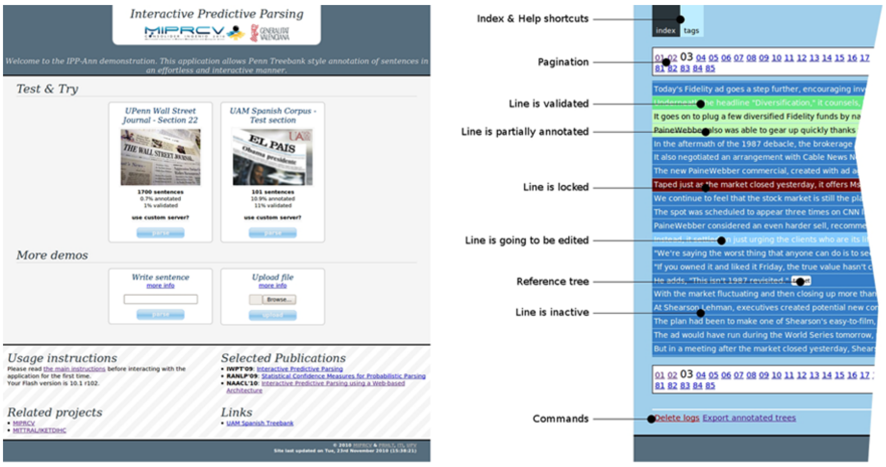
Parsing, also known as grammatical or syntactic analysis, is considered a fundamental problem in Computational Linguistics [10]. Parsing has been also applied to many other research fields aside from the Natural Language Processing (NLP) domain, since the concept of "sentence" can be easily extended to other objects, e.g., mathematical equations. Having perfectly annotated syntactic trees allows one to train and improve the statistical models of other NLP systems, such as machine translation, question answering, or discourse analysis, just to name a few. However, until now the grammatical construction of such trees has been done manually, implying thus a really laborious task.

In this section we introduce a multimodal prototype that operates over the WWW, which is based on the Interactive Parsing framework (IP from here onwards) presented in Sect. 9.2. It can be accessed online at http://cat.iti.upv.es/ipp/. In the prototype, user feedback is provided by means of traditional keyboard/mouse operations and, if available, pen strokes. Interaction through the user feedback allows to improve system accuracy, while multimodality increases both system ergonomy and user acceptability. Furthermore, owing to its web-based nature, this system allows for collaboration between several users across the globe.

12.6.1 Prototype Description

The prototype coordinates client-side scripting with server-side technologies, by using the CAT-API library [1]. The system architecture is based on the CAT network set-up (e.g. see Fig. 12.1). To begin, the user interface (see Fig. 12.17) loads an index of all available corpora. The user then navigates to the chosen corpus page and starts parsing the sentences one by one. She can make corrections both with the keyboard and the computer mouse. User's feedback is then processed by the IP server.

Multimodality is implemented in such a way that the interface accepts two types of human interaction, either by using the keyboard (a deterministic signal) and/or the mouse. The latter, in turn, provides two kinds of operations: *mouse actions* (i.e. clicks, draw movements) and *mouse feedback* (handwritten gestures)—a non-deterministic signal that needs to be decoded by an on-line HTR system, such as the subsystem depicted in Sect. 12.2.1. Both keyboard actions and mouse feedback are used mainly to modify the label of parsing tree's constituents. On the other hand, mouse actions are used to change the span of the above-mentioned tree's constituents (e.g. by clicking on a tree leaf the span is decremented; by drawing a line from a constituent to a word token the span is incremented).



Interactive Predictive Parsing

Welcome to the IPP-Ann demonstration. This application allows Penn Treebank style annotation of sentences in an effortless and interactive manner.

Test & Try

UPenn Wall Street Journal - Section 22
1700 sentences
0.7% annotated
1% validated
use custom server?

UAM Spanish Corpus - Test section
101 sentences
10.9% annotated
11% validated
use custom server?

More demos

Write sentence

Upload file

Usage instructions
Please read the main instructions before interacting with the application for the first time.
Your Flash version is 10.1 r102.

Selected Publications
- IWPT'09: Interactive Predictive Parsing
- RANLP'09: Statistical Confidence Measures for Probabilistic Parsing
- NAACL'10: Interactive Predictive Parsing using a Web-based Architecture

Related projects
- MIPRCV
- MITTRAL/IKETDIHC

Links
- UAM Spanish Treebank

© 2010 MIPRCV & PRHLT, ITI, UPV
Site last updated on Tue, 23rd November 2010 (15:38:21)

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
81 82 83 84 85

Today's Fidelity ad goes a step further, encouraging inv
Underneath the headline "Diversification," it counsels,
It goes on to plug a few diversified Fidelity funds by na
PaineWebber also was able to gear up quickly thanks
In the aftermath of the 1987 debacle, the brokerage
It also negotiated an arrangement with Cable News N
The new PaineWebber commercial, created with ad a
Taped just as the market closed yesterday, it offers Ms
We continue to feel that the stock market is still the pla
The spot was scheduled to appear three times on CNN
PaineWebber considered an even harder sell, recomme
Instead, it settled on just urging the clients who are its li
"We're saying the worst thing that anyone can do is to se
"If you owned it and liked it Friday, the true value hasn't c
He adds, "This isn't 1987 revisited."
With the market fluctuating and then closing up more than
At Shearson Lehman, executives created potential new co
The plan had been to make one of Shearson's easy-to-film,
The ad would have run during the World Series tomorrow,
But in a meeting after the market closed yesterday, Shear

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17
81 82 83 84 85

(a) Index page **(b)** Interface parts

Fig. 12.17 MM-IP layout description

Predictive interaction is approached in such a way that both the main and the feedback data streams help each other to optimize overall performance and usability. The first data stream relates to the validated subtree structure itself. The latter data stream takes into account the provided user information. As already mentioned, since the users operate within a web browser, the system also provides cross-platform compatibility and requires neither computational power nor disk space on the client machine. the client machine. See Fig. 12.18 for a layout of the GUI of this prototype.

Each validated user interaction is saved as a log file on the server side. In this way, the user can retake the tree from its latest state, also allowing other experts to help in parsing the full corpus.

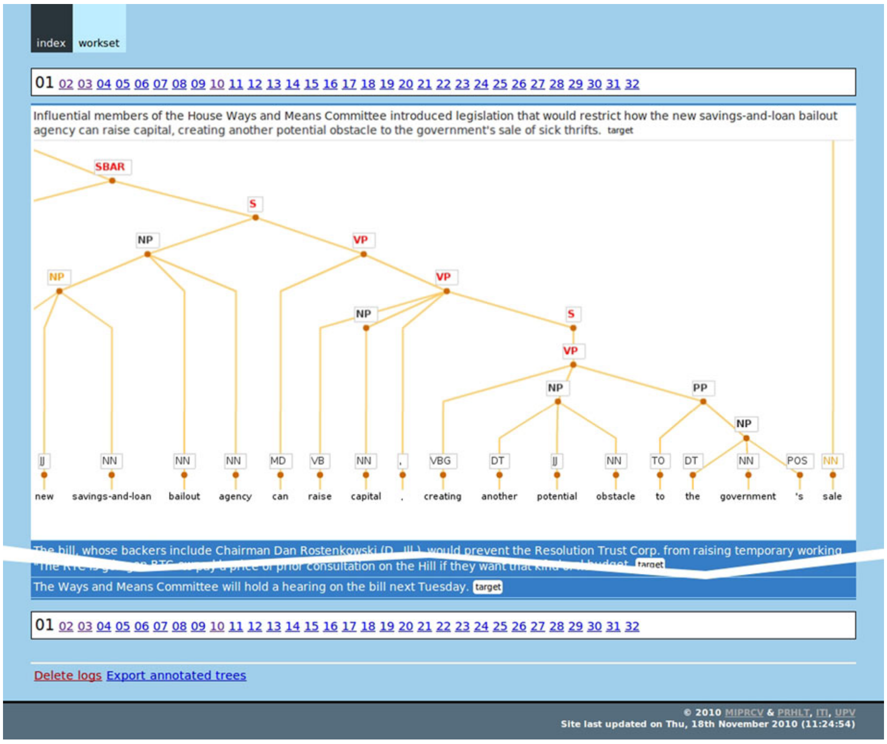
User Interaction Protocol

Each parse tree t is composed by several constituents c_i. Each constituent is comprised of a label, either a *syntactic label* or a *Part-of-speech (POS) tag*, and a span (the starting and ending indexes which delimit the input sentence substring encompassed by the constituent).

The protocol that rules this process is formulated in the following steps.

1. The parsing server proposes a full parse tree t for the input sentence. The tree t is shown on the screen by the client interface.
2. The user finds the longest error-free prefix tree[2] and starts amending the first (incorrect) constituent c from the remaining suffix (either by entering on-line

[2]The tree visiting order is left-to-right depth-first.


index
workset
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
SBAR
S
NP
VP
NP
VP
NP
S
VP
NP
PP
NP
JJ NN NN NN MD VB NN , VBG DT JJ NN TO DT NN POS NN
new savings-and-loan bailout agency can raise capital , creating another potential obstacle to the government 's sale

Influential members of the House Ways and Means Committee introduced legislation that would restrict how the new savings-and-loan bailout agency can raise capital, creating another potential obstacle to the government's sale of sick thrifts. target

The bill, whose backers include Chairman Dan Rostenkowski (D., Ill.), would prevent the Resolution Trust Corp. from raising temporary working

"The RTC is going to have to pay a price of prior consultation on the Hill if they want that kind of budget." target

The Ways and Means Committee will hold a hearing on the bill next Tuesday. target

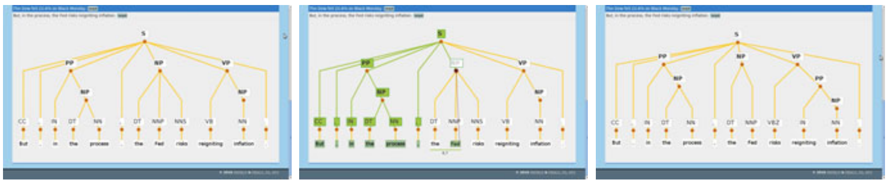
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
Delete logs Export annotated trees
© 2010 MIPRCV & PRHLT, ITI, UPV
Site last updated on Thu, 18th November 2010 (11:24:54)

Fig. 12.18 MM-IP User Interface. Parse trees can be zoomed and/or panned for easing the human annotator's work

touchscreen pen-strokes or amendment keyboard strokes). This operation implicitly validates the prefix tree t_p.

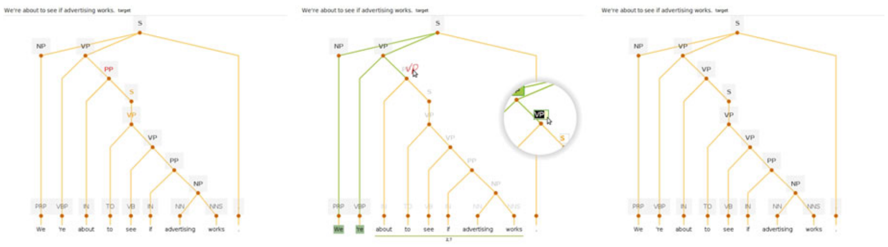
3. The system decodes the user feedback, that is, the mouse/touchscreen pen-strokes or keyboard strokes. This user feedback can either affect the label or the span of the incorrect constituent c:
   (a) If the span of c is modified, the label is assumed to be incorrect. A partial constituent c^*, which includes *span* but no *label*, is decoded from the user feedback.
   (b) If the label of c is modified, the span is assumed to be correct. The corrected constituent c' is decoded from the user feedback.

   This step only deals with analyzing the user feedback, as the parsing server will not be contacted until the next step.
4. Either the partially corrected constituent c^* or the corrected constituent c' is then used by the client to create a new *extended consolidated prefix* that combines the validated prefix and the user feedback: either t_pc^* or t_pc'. The client sends the extended prefix tree to the parsing server and requests a suitable continuation for the parse tree, or tree suffix t_s:



(a) System: Output tree 1 **(b)** User: Span modification **(c)** System: Output tree 2

Fig. 12.19 Mouse interaction example on the IP prototype. A constituent gets its span decremented using the mouse



(a) System hypothesis **(b)** User feedback **(c)** System response

Fig. 12.20 Multimodal interaction example on the IP prototype. Confidence measures (color coding of constituents in the first subfigure) can guide the user toward a faster error detection. Correcting tree constituents can be performed by means of natural pen strokes or keystrokes (*enclosed circle* in (**b**)), leading to a new predicted tree

  (a) If the extended prefix is partial ($t_p c^*$), the parsing server produces a suitable tree suffix t_s. The first element of t_s is the label completing c^*, followed by the remaining calculated constituents.
  (b) If the extended prefix is complete ($t_p c'$), the parsing server produces a suitable tree suffix t_s which contains the remaining calculated constituents.

5. These previous steps are iterated until a final, perfect parse tree is produced.

Notice the similarities with the user simulation subsystem introduced in Sect. 9.5. Within this protocol, tree constituents can be deleted or inserted by adequately modifying the span of the left-neighboring constituent. In Figs. 12.19 and 12.20 one can see live interaction examples.

12.6.2 Technology

Parsing System

A customized CYK-Viterbi parser together with a grammar model is used as the parse server. We made available our system with two grammars, one for the English language constructed using roughly 40 000 sentences from the Penn Treebank, and one for the Spanish language constructed using 1 400 sentences from the UAM Treebank. Notice that the system performed well even with a grammar derived from a small number of sentences, like the one corresponding to the UAM corpus. Refer to Sect. 9.5 for more details on the implementation and grammar construction.

Web Interface

Client–Web server communication is based on asynchronous HTTP connections, providing thus a richer interactive experience—no page refreshes are required, only for changing the corpus to parse. The web server communicates with the IP engine through binary TCP sockets. Thus, response times are adequate. If mouse feedback is used to correct the label of constituents, an on-line HTR decodes the signal and passes the information back to the IP engine. Additionally, cross-domain requests are possible. So, the user can switch between different IP engines from the same user interface.

12.6.3 Evaluation

As mentioned on Sect. 9.5, it is expected that users employing this system would save around 40% of effort, compared to manually post-editing the output of the baseline system.

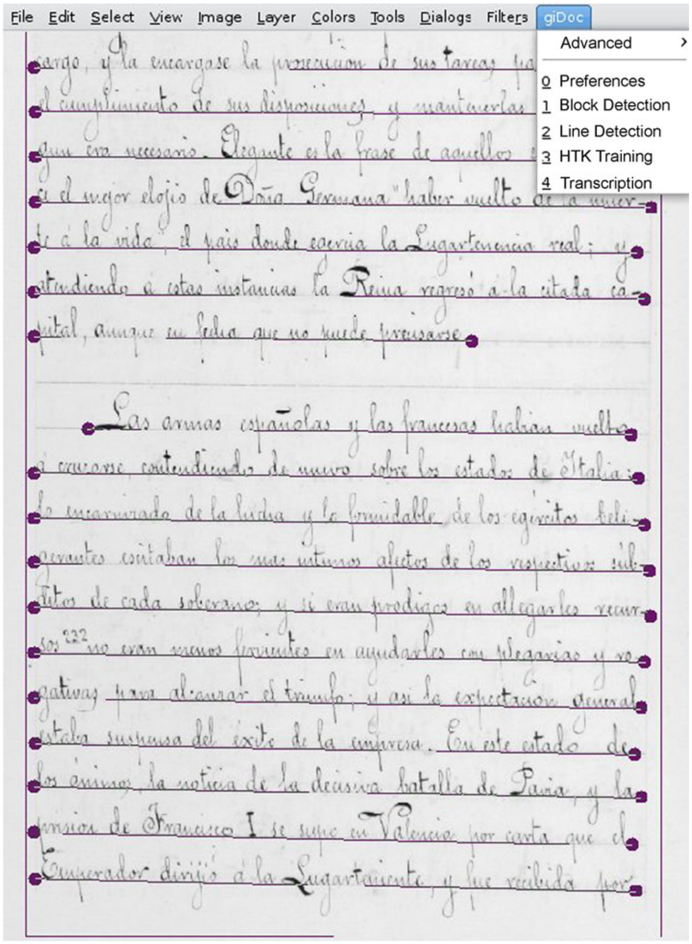
12.7 GIDOC: GIMP-Based Interactive Document Transcription

State-of-the-art technologies for automatic text transcription [17] can hardly deal with handwritten text or even with old-style printed text.

In this section we introduce a demonstrator of the approach presented in Chap. 5. GIDOC (Gimp-based Interactive transcription of text DOCuments) is a computer-assisted transcription prototype [26] for handwritten text images. It is a first attempt to provide integrated support for interactive-predictive page layout analysis, text line detection, and handwritten text transcription [18, 25]. GIDOC is built on top of the well-known GNU Image Manipulation Program (GIMP), and uses standard techniques and tools for handwritten text preprocessing and feature extraction, HMM-based image modeling, and language modeling.

12.7.1 Prototype Description

We decided not to build this prototype from scratch, but on top of GIMP, since GIMP gives us for free many desired prototype features. In particular, GIMP is Free



cargo, y la encargase la procuracion de sus tareas ya
el cumplimiento de sus disposiciones, y mantenerlas
que era necesario. Elegante es la frase de aquellos
el mejor elogio de Doña Germana haber vuelto de a nuev-
o á la vida el pais donde ejercia la Lugartenencia real; y
atendiendo á estas instancias la Reina regresó á la citada ca-
pital, aunque en fecha que no puede precisarse.

Las armas españolas y las francesas habian vuelto
á cruzarse contendiendo de nuevo sobre los estados de Italia;
lo encarnizado de la lucha y lo formidable de los ejércitos beli-
gerantes excitaban los mas intimos afectos de los respectivos súb-
ditos de cada soberano; y si eran pródigos en allegarles recur-
sos 222 no eran menos fervientes en ayudarles con plegarias y ro-
gativas para alcanzar el triunfo; y así la expectacion general
estaba suspensa del éxito de la empresa. En este estado de
los ánimos la noticia de la decisiva batalla de Pavia, y la
prision de Francisco I se supo en Valencia por carta que el
Emperador dirigió á la Lugarteniente, y fue recibida por

Fig. 12.21 Sample image window under the GIDOC menu

Software (GPL), multi-platform, multilingual, and provides: (1) a high-end user interface for image manipulation, (2) a large collection of image conversion drivers and low-level processing routines, (3) a scripting language to automate repetitive tasks, and, specially among others, (4) an API for the installation of user-defined plug-ins. Indeed, the GIDOC prototype is implemented as a set of GIMP plug-ins.

GIDOC is designed to work with (large) collections of homogeneous documents, that is, of similar structure and writing styles. They are annotated sequentially, by (partially) supervising hypotheses drawn from statistical models that are constantly updated with an increasing number of available annotated documents. And this is done at different annotation levels. To run GIDOC, one must first run GIMP and open a document image. GIMP will come up with its high-end user interface, which is often configured to only show the main toolbox (with docked dialogs) and an image window. GIDOC can be accessed from the menu-bar of the image window (see Fig. 12.21), by manipulating the six entries of its main menu:

- `Advanced`: a second-level menu where experimental features of GIDOC are grouped.
- `0:Preferences`: opens a dialog to configure global options, as well as more specific options for preprocessing, training and recognition.
- `1:Block Detection`: performs block detection on the current image.
- `2:Line Detection`: detects (straight) text baselines in the current block.
- `3:Training`: reads all text line images that have been already transcribed and trains from them statistical models for prediction of transcriptions.
- `4:Transcription`: opens the interactive-predictive transcription dialog of GIDOC.

As GIMP, GIDOC is licensed under the GNU General Public License, and it is freely available at http://prhlt.iti.es/w/gidoc. For a detailed review of the prototype's preferences one may see http://prhlt.iti.es/projects/handwritten/idoc/gidoc/manual/.

Block Detection

Detection of (textual) blocks logically precedes text line detection and handwritten text recognition. Technically, textual blocks are represented as closed paths with four handlers at the "corners", which can be graphically adjusted by the user. During its development, GIDOC has been mainly tested on a old book in which most pages only contain nearly calligraphic text written on ruled sheets of well-separated lines, as in the example shown in Fig. 12.21. As mentioned previously, GIDOC is designed to work with such homogeneous documents and, indeed, it takes advantage of their homogeneity. In particular, the Block Detection entry in the GIDOC menu uses a novel text block detection method in which conventional, memoryless techniques are improved with a "history" model of text block positions. An illustrative example is shown in Fig. 12.21, where a blue square encloses the text block.

Line Detection

Given a textual block, the Line Detection entry in the GIDOC menu detects all its text baselines, which are marked as straight paths. The result can be clearly observed in Fig. 12.21 and in the rightmost image of Fig. 12.22. As with the Block Detection, each baseline has handlers to graphically correct its position. Although each baseline has handlers to graphically correct its position, it is worth noting that the baseline detection method implemented works quite well, at least in pages like that of the example Fig. 12.21. It is a rather standard projection-based method [11]. First, horizontally-averaged pixel values or black/white transitions are projected vertically. Then, the resulting vertical histogram is smoothed and analyzed so as to

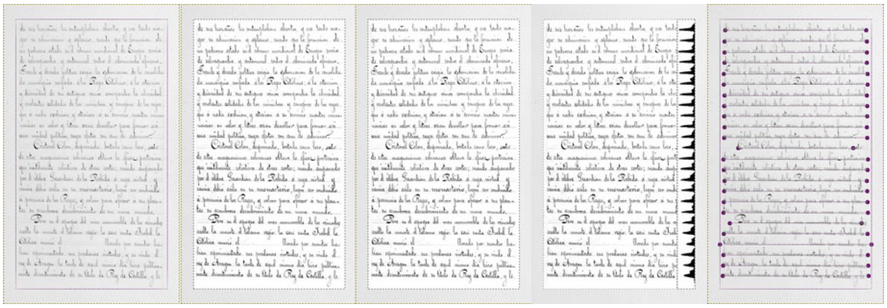


Fig. 12.22 Example of GIDOC Line Detection. *From left to right*: original image, contrast normalization, skew correction, vertical projection of black-to-white transitions, and baseline detection

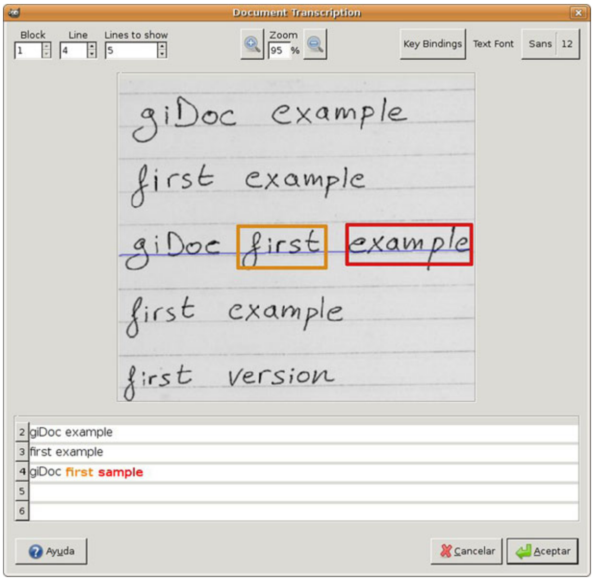
locate baselines accurately. Two preprocessing options are included in `Preferences`; first, to decide on the histogram type (pixel values or black/white transitions), and second, to define the maximum number of baselines to be found. Concretely, this number is used to help the projection-based method in locating (nearly) blank lines.

The line detection method implemented in GIDOC is illustrated in Fig. 12.22. The original image is normalized in contrast, skew-corrected, and then a vertical projection of black-to-white transitions is computed, from which the vertical positions of baselines are derived.

Training

GIDOC is based on standard techniques and tools for handwritten text preprocessing and feature extraction, HMM-based image modeling, and language modeling [29]. Handwritten text preprocessing applies image denoising, deslanting and vertical size normalization to a given text (line) image. It can be configured through `GIDOC/Preprocessing` option. There is an option to use instead a customized procedure, and two options to define (bounds for) the locations of the upper and lower lines, with respect to the baseline.

HMM image modeling can be carried out with the well-known and freely available Hidden Markov Model ToolKit (HTK) [32], and the open source SRI Language Modeling toolkit (SRILM) [28]; or it can be trained using a free software built-in trainer. Training takes place using examples, therefore at least some pages have to be transcribed previously. As more transcriptions are available, better models are obtained. Hence, training must be called every few pages that have been transcribed. In this way the recognition system's performance is greatly improved. Specifically, two options are available in this process: train from the beginning all the models or re-estimate the current ones.



2 giDoc example
3 first example
4 giDoc first sample
5
6

Fig. 12.23 GIDOC Interactive Transcription dialog

Transcription

The `GIDOC/Transcription` entry in the main menu opens an interactive transcription dialog (see Fig. 12.23). It consists of two main sections: the image section, at the middle part of the window, and the transcription section, at the bottom part. Text line images are displayed in the image section together with their transcriptions, if available, in separate editable text boxes within the transcription section. The current line to be transcribed (or simply supervised) is selected by placing the edit cursor in the appropriate editable box. Its corresponding baseline is emphasized and, whenever possible, GIDOC shifts line images and their transcriptions so as to display the current line in the central part of both the image and transcription sections. It is assumed that the user transcribes or supervises text lines, from top to bottom, by entering text and moving the edit cursor with the arrow keys or the mouse. However, it is possible for the user to choose any specific, desired order.

Each editable text box has a button attached to its left, which is labeled with its corresponding line number. By clicking on it, its associated line image is extracted, preprocessed, transformed into a sequence of feature vectors, and Viterbi-decoded using the HMMs and language models trained with `GIDOC/Training`. The Grammar Scale Factor (GSF) and Word Insertion Penalty (WIP) values to properly combine the HMM and language models are defined in the recognition section of `GIDOC/Preferences`. Also there is an option to adjust the *Beam* value and thus the computational cost to perform Viterbi decoding. In this way, there is not need to enter the complete transcription of the current line, but hopefully only minor corrections to the decoded output. Clearly, this is only possible if, first, text lines are correctly detected and, second, the HMM and language models are adequately trained, from a sufficiently large amount of training data. Therefore, it is assumed that the transcription process is carried out manually in early stages of a transcription task, and then is assisted as described here.

Apart from the image and transcription sections, the dialog shown in Fig. 12.23 includes a number of controls in the top part, as well as self-explanatory buttons under the transcription section. Regarding the controls in the top part, note that they allow the user to select the current block of the image to transcribe, the current line, the number of lines to show, etc.

If verification is enabled, suspicious words are emphasized in orange (neither reliable nor completely unreliable) or red (totally unreliable); see Fig. 12.23. If properly adjusted, verification can assist the user in locating possible transcription errors, and thus validate system output after only supervising those (few) words for which the system is not highly confident. Validation of lines after supervision is done by simply pressing `Enter`. It is worth noting that only validated lines are used to adapt (re-train) the system through `GIDOC/Training`.

User Interaction Protocol

The user follows a step-by-step process to fully transcribe a page, consisting on a given handwritten image. Firstly she opens an image and sets up the project preferences: `Preprocessing`, `Training`, and `Recognition` options. She then defines the text block in the image, by creating a rectangular selection containing the text block. The interactive transcription dialog appears and she transcribes manually a few lines, with the purpose of training the system. The system models are then generated and the user enters in an interactive transcription process, where he/she works on the detected lines one by one. It is worth noting that GIMP offers non-interactive functionality through the command line, so GIDOC can also work in this way. For example, it is possible to extract text lines from a (set of) handwritten image(s), preprocess them, and save the result in any directory.

12.7.2 Technology

As previously commented in Sect. 12.7.1, GIDOC is implemented as a set of GIMP plug-ins, and thus GIMP has to be installed in the first place. In addition, the training and prediction engines are based on the methods described in Chap. 5.

12.7.3 Evaluation

Early testing of the GIDOC prototype with real users revealed a time reduction in whole transcription tasks of 25% on average, in a moderately complex text documents. During its development, GIDOC has been used by a paleography expert to annotate blocks, text lines and transcriptions on a dataset called GERMANA [16]. The example shown in Fig. 12.21 corresponds to page 144 of such a dataset. The reader is redirected to Sect. 5.6 for a detailed evaluation of results based on this prototype.

12.8 RISE: Relevant Image Search Engine

The classical Content Based Image Retrieval (CBIR) approach helps to organize digital pictures by their visual content, and traditionally has involved a myriad of multidisciplinary fields such as computer vision, machine learning, HCI, database systems, statistics, and many more [7, 12, 31]. Two problems arise, though. On the one hand, it is not feasible to make the annotations of all images manually, and, on the other hand, it is not possible to translate some images (specially abstract concepts) into words—which is known as *the semantic gap* [27]. Relevance feedback (RF) is a query modification technique which attempts to capture the user's precise needs through iterative feedback and query refinement [7].

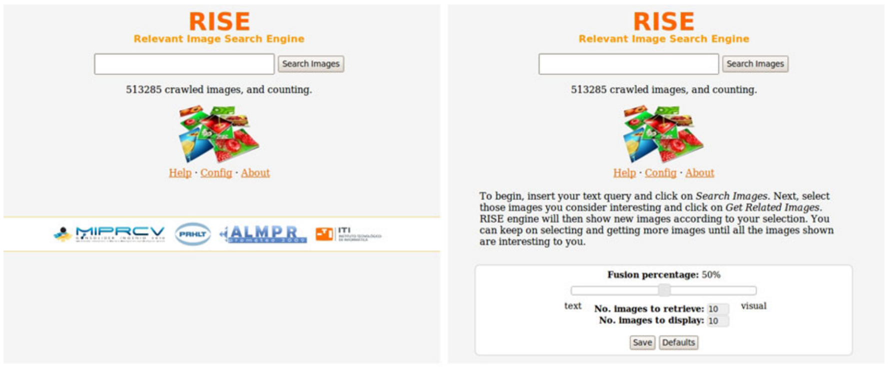
In this section we shall demonstrate the methodology described in Chap. 11, which is implemented in a Relevant Image Search Engine (RISE from here onwards) with late fusion. Fusion is used here to combine two *modalities* used to describe queries: *visual* or image-based and *semantic* or text-based. For some applications, visual similarity may in fact be more critical than semantic, and vice versa. The late fusion concept allows one to blend the amount of textual and visual information that will be used to retrieve similar images, overcoming thus with the inherent problem of the above-mentioned semantic gap. There is an online demonstration available at http://risenet.iti.upv.es/rise/.

12.8.1 Prototype Description

User Interaction Protocol

In sum, the user has in mind some relevant set of (unknown) images, and RISE's goal is to interactively discover n images of it, among the images in a fixed, finite collection of images C: The process is described as follows:

- Initially the user inputs a query q to the system.
- Then RISE provides an initial set $X_0 \in C$ of n images that are retrieved from a database according to a suitable criteria.
- These images are judged by the user, who provides a feedback by selecting which images are relevant (and, implicitly, which are not relevant).
- The system combines such feedback information with late fusion methods to obtain a new set of images X_1 depending of the nature of q, i.e., text-based or visual.
- The process is repeated until the user is satisfied, i.e., all retrieved images X_i are relevant to her.



513285 crawled images, and counting.

To begin, insert your text query and click on *Search Images*. Next, select those images you consider interesting and click on *Get Related Images*. RISE engine will then show new images according to your selection. You can keep on selecting and getting more images until all the images shown are interesting to you.

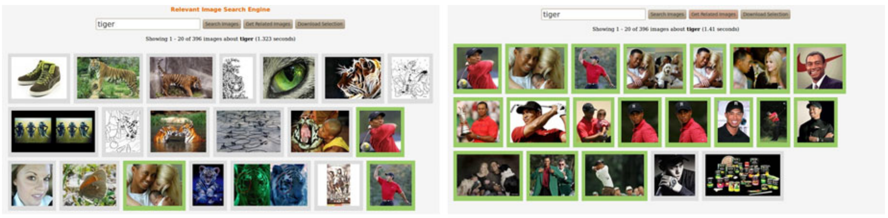
Fusion percentage: 50%

text **No. images to retrieve:** 10 visual
No. images to display: 10

Save Defaults

(a) (b)

Fig. 12.24 RISE index page



Showing 1 - 20 of 396 images about **tiger** (1.323 seconds)

Showing 1 - 20 of 396 images about **tiger** (1.41 seconds)

(a) (b)

Fig. 12.25 Interacting with the RISE prototype. The user typed the (ambiguous) query “tiger”, and the system provides images related to the feline (**a**). However, the user was thinking of the sportsman Tiger Woods, so she validates those images she consider as relevant, and the system will use her feedback for improving their results in the next iterations (**b**)

Web Interface

On index page (Fig. 12.24a) the user can configure the fusion amount between text- and visual-based engines (see Fig. 12.24b). Once the user has submitted a query to the system, a set of images is shown, allowing the user navigate to interactively inspect some image properties such as dimensions, size, of referrer URL (Fig. 12.26).

12.8.2 Technology

We implemented a probabilistic model for user RF on CBIR, detailed in Sect. 11.2 (see also [15]), which is augmented with late fusion techniques. For this demonstrator we considered a simple (yet very effective) score based on a weighted linear combination of N-best lists, i.e., the best ranking of (relevant) images R_b is computed a as linear combination of visual R_v and text R_t rankings: $R_b(\%) =$

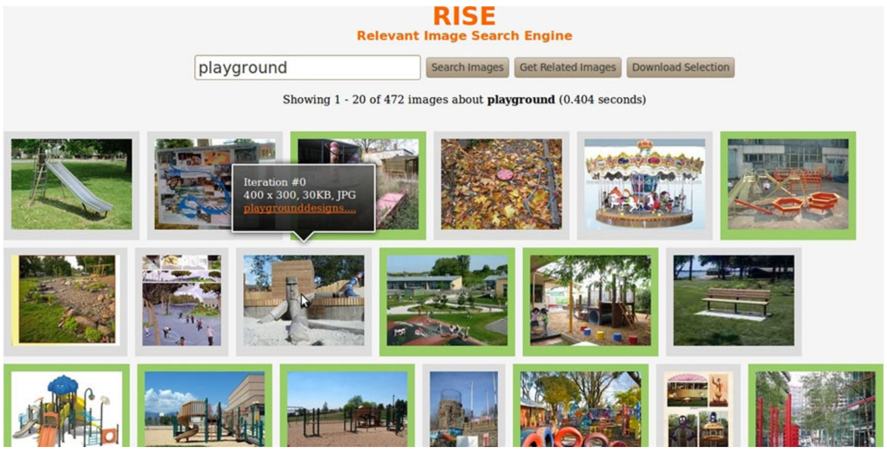


Fig. 12.26 RISE's User Interface. After introducing a text query, the user is presented with a set of images, having to select those she finds relevant (highlighted in *green*). Additional information is displayed when hovering on each image

$\alpha R_v + (100 - \alpha) R_t$, where α is automatically adjusted to account for the fusion percentage between visual and textual retrieval strategies. See details in Sect. 11.3.

Starting from Scratch

As starting point to build the image collection C, we used the Merriam-Webster's online dictionary to generate a list of queries to search for. After shuffling the 35 000 words in such an initial list we performed 1 500 searches for each dictionary entry among the main search engines (namely Google, Bing, and Yahoo) by using the GNU `wget` application (named *image crawler* from here onwards).

Gathering Images

The image crawler stores a thumbnail as well as the feature vectors in an optimized representation for searching, requiring thus minimal storage space; e.g, 1 000 000 retrieved images take up to just 10 GB overall, including all textual descriptors. The context in which an image appears is abstracted from the containing HTML document using the Latent Semantic Indexing (LSI) method. LSI works by statistically associating related words to the conceptual context of the given document. This structure is estimated by a truncated singular value decomposition (SVD) [5].

Automatic Image Tagging

The main code was implemented in Python, with the help of the Beautiful Soup HTML/XML parser library. The first part of the image tagging process consists in deleting all irrelevant elements (such as tables or scripts) from the retrieved web pages. The next step consists of assigning special importance to certain words with specific HTML tags, for example, one may assign different weights to the keywords appearing in the page title, headers, or in the attributes of the `img` tags. We also decided to use some of the related search engine's information. For instance, the closer to the first page of results is an image, the more important it is. A dictionary is thus created with the term frequency of each word that appears in the web document, alongside with the bag of weighted words. Then, and after applying some normalization routines (e.g., filtering stop words, lower-casing terms), image annotations are stored in the database (including their weights).

Retrieval Procedure

Initially we use text queries to narrow the initial set of images that will be presented to the user, i.e., she types in an input field what she is looking for. Then we employ the *query-by-similar-images* paradigm with late fusion in an interactive setting (see Sect. 12.8.1).

12.8.3 Evaluation

The evaluation and experiments are presented in Sect. 11.3.5. In general, we observed that depending on the query, a pure visual strategy may help achieving a complete set of relevant images while, in other cases, pure text retrieval performs better (take for instance the example commented in Fig. 12.25).

This is consistent with the performance results reported in Chap. 11, which compares the accuracy for the best α with both pure text and pure visual retrieval.

12.9 Conclusions

In this chapter we have introduced several demonstrators of multimodal interactive pattern recognition applications. These demos have served as validating examples of the MIPR framework proposed and described throughout this book. As observed, they involved many different real-world tasks, namely transcription of text images and speech, machine translation, text generation, parsing and image retrieval. Each of the seven introduced prototypes was based on one of the three types of user interaction protocols, and tapped the user feedback to enrich the system output. The most remarkable features are summarized as follows.

First, multimodal interaction has been approached in such a way that both the main and the feedback data streams collaborate to optimize the overall performance and usability. The interaction process is iterated until the user considers that the system output is wholly correct, so a perfect and high-quality output is guaranteed, since the user is tightly embodied on the system's inner processing. Second, an overview of each demonstrator has been sufficiently detailed to give the reader an overview of the underlying technologies. Most of the above-mentioned prototypes are being currently adapted to work as online demos, therefore being able to expand the potential audience. Third, all systems have been submitted to preliminary evaluation with real users. The resulting figures are definitely promising, validating thus the interactive–predictive paradigm off the lab. Finally, the reader is encouraged to test the available prototypes (e.g., the web-based versions), since the requirements on the client side are really low, and check the features that have been put into practice.

References

1. Alabau, V., Romero, V., Ortiz-Martínez, D., & Ocampo, J. (2009). A multimodal predictive-interactive application for computer assisted transcription and translation. In *Proceedings of international conference on multimodal interfaces (ICMI)* (pp. 227–228).
2. Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A. L., Ney, H., Tomás, J., Vidal, E., & Vilar, J. M. (2009). Statistical approaches to computer-assisted translation. *Computational Linguistics*, *35*(1), 3–28.
3. Bickel, S., Haider, P., & Scheffer, T. (2005). Predicting sentences using n-gram language models. In *Proceedings of human language technology and empirical methods in natural language processing (HLT/EMNLP)* (pp. 193–200).
4. Bisani, M., & Ney, H. (2004). Bootstrap estimates for confidence intervals in ASR performance evaluation. In *Proc. ICASSP* (pp. 409–412).
5. Cascia, M. L., Sethi, S., & Sclaroff, S. (1998). Combining textual and visual cues for content-based image retrieval on the world wide web. In *IEEE workshop on content-based access of image and video libraries* (pp. 24–28).
6. Craciunescu, O., Gerding-Salas, C., & Stringer-O'Keeffe, S. (2004). Machine translation and computer-assisted translation: a new way of translating? *Translation Journal*, *8*(3), 1–16.
7. Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, *40*(2), 1–60.
8. Jelinek, F. (1998). *Statistical methods for speech recognition*. Cambridge: MIT Press.
9. Koehn, P., Och, F. J., & Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the HLT/NAACL* (pp. 48–54).
10. Lease, M., Charniak, E., Johnson, M., & McClosky, D. (2006). A look at parsing and its applications. In *Proc. AAAI* (pp. 1642–1645).
11. Likforman-Sulem, L., Zahour, A., & Taconet, B. (2007). Text line segmentation of historical documents: a survey. *International Journal on Document Analysis and Recognition*, *9*, 123–138.
12. Moran, S. (2009). *Automatic image tagging*. Master's thesis, School of Informatics, University of Edinburgh.
13. Oncina, J. (2009). Optimum algorithm to minimize human interactions in sequential computer assisted pattern recognition. *Pattern Recognition Letters*, *30*(5), 558–563.
14. Ortiz-Martínez, D., Leiva, L. A., Alabau, V., & Casacuberta, F. (2010). Interactive machine translation using a web-based architecture. In *Proceedings of the international conference on intelligent user interfaces* (pp. 423–425).
15. Paredes, R., Deselaer, T., & Vidal, E. (2008). A probabilistic model for user relevance feedback on image retrieval. In *Proceedings of machine learning for multimodal interaction (MLMI)* (pp. 260–271).

16. Pérez, D., Tarazón, L., Serrano, N., Castro, F.-M., Ramos-Terrades, O., & Juan, A. (2009). The GERMANA database. In *Proceedings of the international conference on document analysis and recognition (ICDAR)* (pp. 301–305).
17. Plötz, T., & Fink, G. A. (2009). Markov models for offline handwriting recognition: a survey. *International Journal on Document Analysis and Recognition*, *12*(4), 269–298.
18. Ramos-Terrades, O., Serrano, N., Gordó, A., Valveny, E., & Juan, A. (2010). Interactive-predictive detection of handwritten text blocks. In *Document recognition and retrieval XVII (Proc. of SPIE-IS&T electronic imaging)* (pp. 219–222).
19. Rodríguez, L., Casacuberta, F., & Vidal, E. (2007). Computer assisted transcription of speech. In *Proceedings of the Iberian conference on pattern recognition and image analysis* (pp. 241–248).
20. Romero, V., Toselli, A. H., Civera, J., & Vidal, E. (2008). Improvements in the computer assisted transciption system of handwritten text images. In *Proceedings of workshop on pattern recognition in information system (PRIS)* (pp. 103–112).
21. Romero, V., Leiva, L. A., Toselli, A. H., & Vidal, E. (2009). Interactive multimodal transcription of text images using a web-based demo system. In *Proceedings of the international conference on intelligent user interfaces* (pp. 477–478).
22. Romero, V., Leiva, L. A., Alabau, V., Toselli, A. H., & Vidal, E. (2009). A web-based demo to interactive multimodal transcription of historic text images. In *LNCS: Vol. 5714. Proceedings of the European conference on digital libraries (ECDL)* (pp. 459–460).
23. Sánchez-Sáez, R., Leiva, L. A., Sánchez, J. A., & Benedí, J. M. (2010). Interactive predictive parsing using a web-based architecture. In *Proceedings of NAACL* (pp. 37–40).
24. Sanchis-Trilles, G., Ortiz-Martínez, D., Civera, J., Casacuberta, F., Vidal, E., & Hoang, H. (2008). Improving interactive machine translation via mouse actions. In *EMNLP 2008: conference on empirical methods in natural language processing*.
25. Serrano, N., Pérez, D., Sanchis, A., & Juan, A. (2009). Adaptation from partially supervised handwritten text transcriptions. In *Proceedings of the 11th international conference on multimodal interfaces and the 6th workshop on machine learning for multimodal interaction (ICMI-MLMI)* (pp. 289–292).
26. Serrano, N., Tarazón, L., Perez, D., Ramos-Terrades, O., & Juan, A. (2010). The GIDOC prototype. In *Proceedings of the 10th international workshop on pattern recognition in information systems (PRIS 2010)* (pp. 82–89).
27. Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(12), 1349–1380.
28. Stolcke, A. (2002). SRILM—an extensible language modeling toolkit. In *Proceedings of the international conference on spoken language processing (ICSLP)* (pp. 901–904).
29. Toselli, A. H., Juan, A., Keysers, D., González, J., Salvador, I., Ney, H., Vidal, E., & Casacuberta, F. (2004). Integrated handwriting recognition and interpretation using finite-state models. *International Journal of Pattern Recognition and Artificial Intelligence*, *18*(4), 519–539.
30. Trost, H., Matiasek, J., & Baroni, M. (2005). The language component of the fasty text prediction system. *Applied Artificial Intelligence*, *19*(8), 743–781.
31. Wang, J. Z., Boujemaa, N., Bimbo, A. D., Geman, D., Hauptmann, A. G., & Tešić, J. (2006). Diversity in multimedia information retrieval research. In *Proceedings of the 8th ACM international workshop on multimedia information retrieval* (pp. 5–12).
32. Young, S., et al. (1995). *The HTK book*. Cambridge University, Engineering Department.