USING
NATURAL LANGUAGE PROCESSING
TO ASSIST THE VISUALLY HANDICAPPED
IN WRITING COMPOSITIONS

Jacques Chelin

A Thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

July 2006

Your file Votre référence
ISBN: 978-0-494-34587-0
Our file Notre référence
ISBN: 978-0-494-34587-0

ABSTRACT

Using Natural Language Processing To Assist The Visually Handicapped In Writing Compositions

Jacques Chelin

Over the last decades, the visually handicapped have consistently progressed in their efforts towards inclusion in the mainstream. Integration in education and professional life in particular has become possible due to computers without which it would not have been possible to the same degree. In the wake of this integration, more and more blind students are attempting CEGEP and University studies.

This situation has created new problems and new needs. One of them is the need to study text and electronic documents in depth and in a reasonable time. Blind students cannot flip through the pages of a book, skim through the text or use a highlighter. Natural Language Processing (NLP) is about understanding and processing language as used by humans and includes a subdomain called Information Retrieval. In this thesis, we will describe this problem in detail and review how it has been addressed so far in the industry. We will then propose a solution in the form of an experimental prototype and show how some existing NLP techniques can profitably assist blind students in meeting their academic objectives. This thesis is a technology application and engineering of a practically usable software product.

Acknowledgements

I would like to express my deepest gratitude to my supervisors. For years, **Dr. Radhakrishnan** has had a lot of interest in the use of computers by the visually handicapped to face successfully the challenges they meet when trying to reach their full potential. He was personally involved with the Montreal Association for the Blind for some time. He is the one who decisively influenced me to start my Certificate in User Interfaces which I converted into a Masters and encouraged me all along to continue in my project.

Dr. Kosseim was also of great help in goading me forward, reassuring me in what I was doing and providing valuable insights from the field of NLP.

The work described in this thesis would not have been possible without the contribution of **Dr. Leo Bissonnette** who did not hesitate to delve into his rich experience as an educator and pedagogue when evaluating the system.

My sincere thanks go to them all.

This thesis is dedicated to my family who has inspired, supported and encouraged me for many years, right from the start of my part-time studies. They were present at each step of the way and the merit of this thesis is definitely partly theirs.

Table of contents

List of figures

List of tables

Chapter 1

1. A new problem

1.1. Introduction

The purpose of this thesis is to propose a solution to a socially relevant new problem in using existing technologies. The visually handicapped have consistently progressed over the last decades in their efforts towards inclusion in the mainstream. Integration in education and professional life in particular was possible due to the invention of computers without which it would not have been possible, or at least not to the same degree. In the wake of this integration, more and more blind students are attempting CEGEP and University studies.

This situation has created new problems and new needs specifically related to the in depth study of documents in a reasonable time so as to produce assignment submissions and research papers. Natural Language Processing (NLP) is about understanding and processing language as used by humans. In this thesis, we will investigate how existing NLP techniques can assist blind students in meeting their academic objectives and integrating better into this world of Internet.

1.2. Definition of the problem

On a regular basis, CEGEP and University students, especially those in Liberal Arts, have to do research on specific topics. Their task consists of consulting a wide variety of books, documents and Web sites and producing a document, anything from an assignment to a 10 page essay, a take-home exam or a full-fledged thesis. To be able to understand more precisely the difficulties that the visually handicapped students face when they try to write a paper, we need to give a quick description of the tools they use to read and write.

1.2.1. Reading and writing tools for the blind

The visual handicap can be divided into two broad categories, the partially sighted and the totally blind. We only address the latter here. One way to make up for the absence of sight is the use of speech. The way computer technology is used in this case is to provide spoken output as screen readers that use an internal software speech synthesizer and the computer's sound card. Words appearing on the screen are read aloud. Most of the time, the speech output follows the position of the cursor. The major problem with using speech as a medium is that it provides a very small working window due to the constraints of short term memory and linearity of speech. George Miller[1] (1956) provided

[1] http://www.cogsci.princeton.edu/%7Egeo/

evidence that people can remember about 7 chunks (in our case, words) in short-term memory[2]. Cowan (2001) goes even further and suggested as little as 3 to 5 words.

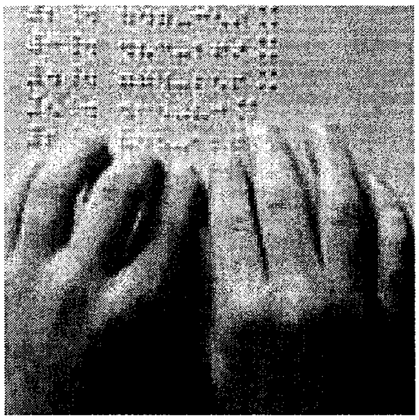

Figure 1.1 – Braille dots as they appear on paper

The other way to compensate for the loss of visual ability is through the sense of touch. For two centuries, the blind have been able to read by moving their fingers across raised dots on thick paper (Figure 1.1)[3]. The dots are organized in cells or a matrix of 3 rows and two columns. Each dot can be raised or left flat. A cell corresponds to a character.

Figure 1.2 should immediately bring home to the reader how the presence and configuration of the dots are used to represent the different characters[4]. Since there are six positions in the cell that may contain a raised or flat dot (a binary combination), 2 to the power of 6 combinations can be used to encode the symbols[5].

---

[2] http://www.well.com/user/smalin/miller.html
[3] Borrowed from http://www.nbp.org/
[4] This figure was realized by using the Braille font available in Microsoft Word 2003.
[5] Additional information can be found at http://www.brailleauthority.org/def.html

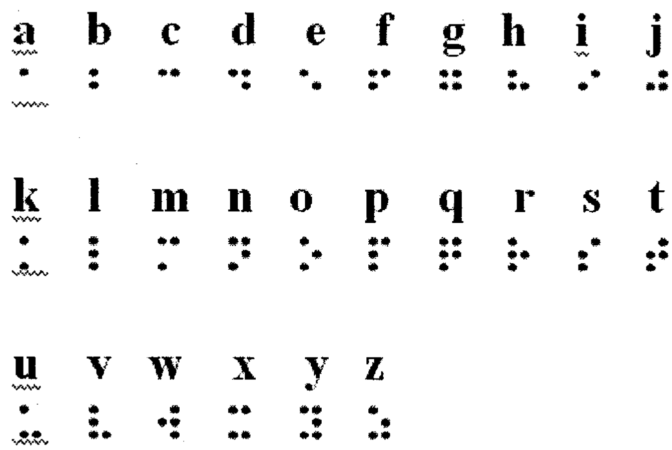

Figure 1.2 – Braille Alphabet

The appearance of computer based 'Braille displays' in which each dot can be raised or lowered through electromechanical devices has caused a huge leap in the accessibility to electronic documents.

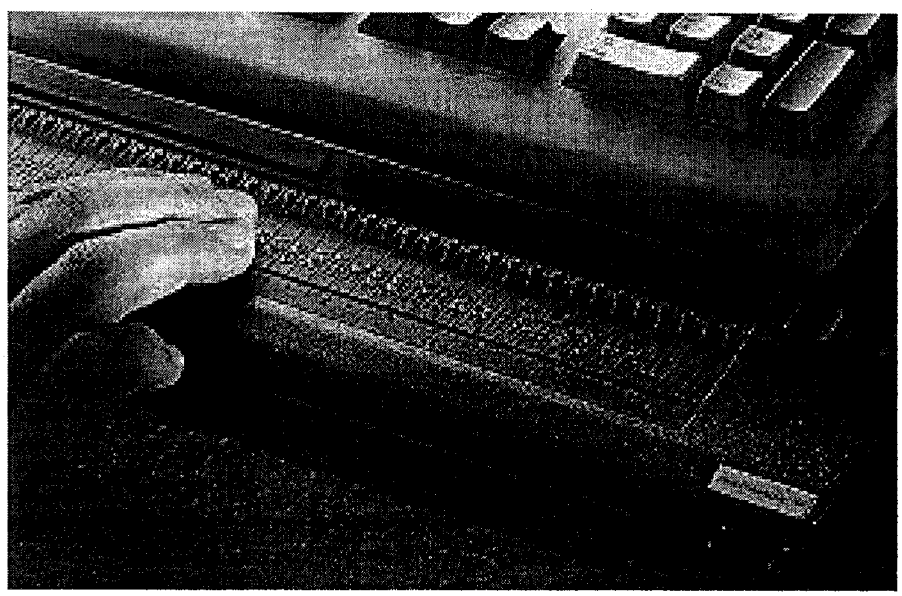

Figure 1.3 – Braille display close-up showing Braille cells[6]

[6] Created from an original image borrowed from http://www.freedomofspeech.com/powerbraille80.html

In a Braille display, the raised dots are achieved by nylon or metal rods that move up and down. The result is that a cell can show any Braille character. Displays come in three formats, 20, 40 or 80 cells forming one line of ‘refreshable’ Braille. 80-cell displays are expensive. Thus the most common are 40-cell displays. In fact, modern displays have cells of 8 dots as can be seen in Figure 1.3. This provides for 2 to the power of 8 configurations and the possibility to represent a richer set of characters or the position of a cursor in the line, a must for computer applications. As one would guess, Braille displays are connected to a computer output port. Screen readers can also output to Braille displays.

The major constraint here is that the user can only ‘see’ a 40-character window. To understand this, the reader would have to imagine that, instead of a full screen of information with several applications opened and immediately visible, the screen would be shut down and the only way he could access his computer would be through the screen of his cellular phone which shows a maximum of 40 characters or through a friend reading out the screen to him.

1.2.2. A comparison

To write a paper, a student normally:

1. Scans a substantial amount of books and documents quickly and easily using fast

reading techniques.

2. Develops an outline while reading one or more of these documents more systematically, “highlighting” and taking notes.
3. Reviews and rearranges his notes and uses them to create a draft of his document.
4. Refines the document iteratively until it is complete.
5. At all times, refers back to previous readings and versions.

At first sight, all these steps seem easy, mundane and as day to day tasks. In real life, several categories of problems exist:

1. The amount of material to be read is, by itself, often a challenge even for a sighted student.
2. For handicapped students, the reading difficulties go from very mild to very severe due to lack of reading and summarization skills, visual handicaps, dyslexia, etc...
3. Note taking is an art in itself. The classic index card method facilitates sorting of notes but, based on our experience, is tedious and hardly used by sighted students. Taking notes can present additional challenges to students with certain handicaps.
4. As shown in section 1.2.1, everything is always seen through a 40 character window for Braille users, or a window of less than 10 spoken words for speech users. Contrary to sighted users who can see a full screen of information, this space restriction creates a major challenge, that of not having any overview at any level.

The above problems are encountered while performing four major tasks: Research, Analysis, Outlining and Composition.

Research:

When starting a paper or a research project, often the student has to get acquainted with a large body of text. At this stage, skimming is a crucial activity study skill. Skimming is impossible for a blind person. What appears to be substantial for a regular student, in terms of the amount of material to be read, becomes massive for a visually handicapped student. Speed is also an issue here. For that reason, fast reading using speech output is required in the beginning of the research stage as a substitute for skimming. This generates the need for multiple readings where one would be sufficient for a sighted reader.

Analysis:

After completing the research task, the student finds herself with a subset of the material that will require more careful study. In this step, highlighting, underlining and note taking are required. Since highlighting and underlining are not possible, extensive notes have to be taken. However, it is difficult to do so in voice mode since it is a drag to stop and start the voice output; this slows down the process too much. Manual note cards are out of the question. The only substitute is cutting and pasting to form an auxiliary document. Since that is too difficult in voice mode and very few or no notes were taken in previous hearings of the material, the visually handicapped students must use Braille to take notes on what he thinks is pertinent in still another pass through the material.

Outlining:

Putting order and structure in the notes is a tedious process. There is no way to structure them easily. Sometimes several physical documents have to be manipulated. If material has to be reviewed again and again based on a reference made in the notes, a book or document may have to be practically “read” several times.

Composition:

This is the easiest part to support, since human intelligence and creativity are in play.

1.3. A practical example – SCENARIO 1

A practical example is depicted in the following scenario[7].

Julian is a blind first year College student. He is just back from the CEGEP and he is all excited. His teacher has just accepted that the subject of his essay be ‘The Crusades’. His second semester includes a course on Methodology where he has to do a big 20-page paper showing he has mastered the art of writing an essay of some consequence. It is the eighth of February and he has until April the 30th to complete it. He has hardly removed his winter coat that he is already making arrangements on the phone with someone to go to the library on the next day to choose his books. He estimates four good books will be enough.

---

[7] This is a real life scenario based on the personal experience of a blind CEGEP student.

Three weeks later, his books have been electronically scanned into Word documents. His first task is to try and have a first grasp of the subject matter. He needs to know what to talk about. He cannot skim through the material to do so. He has to do a first reading of all four books before he can start taking notes. All he can do at this point in time is a fast, cursive reading of the material without taking notes. Of course, his sighted peers may find it helpful to highlight some passages so as to render note taking easier and more efficient. Julian can only rely on his memory. The highlighting facility in Word is not accessible to him.

This takes him another three weeks to read everything because he takes twice as much time as his sighted peers. Now he has a better idea of what to talk about. The time has come to take notes. However, he has partially forgotten what he read previously. He cannot skim or page through a book containing highlights and post-it's like sighted students. So he has to reread the material again. In so doing, he studiously takes notes as he has been told to do so. He summarizes a paragraph and takes note of the page number. This he can use as a quote or an example. He has to save somewhere in another file three paragraphs from page 252. He has to select the text. Having only a 40-character window to work with, this takes time since he often loses his selection and has to start again. The larger the text to copy, the more difficult it is. Doing this exercise with four books takes him to the end of April. He is currently doing his outline that should have been handed in March. He is already lagging behind one month. So he will have to ask for an extension. The others are already turning in their essay.

As he reads some of his material, he realizes that he skipped some things when taking notes. He has to do more reading and more note taking. Eventually the outline is done. He hands it in on the 5th of May. Normally one month is allowed for reviewing and improving the outline based on the teacher's suggestions. This involves additional readings and rework. Two additional books have to be scanned into Word documents. He is relieved to know that this can be done within a week.

Around the end of May, the time comes to do his essay. He finds himself with 10 to 15 files of notes that were used to produce his outline. He could reorganize them. But he finds he needs to go back to the sources to clarify or reformulate some ideas. So, he must use the original books instead of trying to find his way around in those files. He finds them impossible to manage; they are useless to write the essay. He cannot derive from his notes the advantages he expected. In addition, he has two other books to contend with. He rereads the four books and keeps them in front of him to do his essay. They are big files. He must find the places he wanted to quote. More cut and paste are needed. Finally, on the 27th of June, two months after the deadline, he hands in his paper.

1.4. Solution requirements

To solve the problems implied through the above scenario, we need an **Information Probing and Gathering** environment that drastically reduces the time it takes a student to do Research on a specific topic and produce a paper and help him meet the following objectives:

- Determine the relevance of documents without having to read them entirely.
- Access information related to a user specific theme or subject much faster than traditional methods.
- Increase substantially both the amount of information accessed and the speed of access.
- Free the handicapped student from tedious searches and reading chores for more productive and creative work.
- Organize the user notes with "intelligent bookmarks" and instantaneous references to previously processed texts.

More specifically, this requires a **paradigm shift** from the classical and standard document processing application. The purpose here is to investigate where and how far NLP techniques can facilitate the access of students to information pertinent to their research. These techniques will convey additional information about a document consisting of pointers to where the information is relevant. Moreover, the structure of the output document in our specific context is strongly outline-driven. So are the input documents since they are not just read but studied and that type of detailed document analysis calls for close and renewed attention to the structure of the input document. Let us understand why.

1.4.1. A model

To really understand the process of writing a paper, one has to clearly identify the ultimate goal. Let us view the ultimate document produced to consist of three elements, a table of contents which is derived from the final outline, a body of text written by the student and quotations from other authors. The process to generate each of these involves assimilating and filtering a number of texts around an outline which has a very important pivotal role. This process is significantly iterative. Preliminary material is often read to derive an outline which will then be used to direct the research using the selected input material. This research will bring changes and additions to the outline requiring further research on more selective material. This process continues until a complete outline is made that we view as a structured set of texts and quotations.

1.4.2. Creating the outline

The very first goal for a writer is to develop a set of key terminology and a rough outline. Thus, the first phase involves gathering a minimum of information so as to meet this objective.

> *"Reading and note taking must relate directly to specific needs expressed as key terminology or a rough outline... You need to know what to look for and why you need it".*

[Lester, 1996 p. 81]

"(This) outline, although sketchy, provides the terminology for scanning sources, checking alphabetic indexes, and conducting interviews or questionnaires"

[Lester, 1996 p. 83]

Lester (1996) proposes the following tasks to reach this goal:

1. Jot down ideas or words in a rough list.
2. Classify them into a hierarchy of major and minor ideas.
3. Formulate questions to identify issues.
4. Use research proposal or thesis (to chart the direction of the research)
5. Goals can and must be revised in a recursive fashion. Thus key terminology can change, requiring new searches.

1.4.3. Expanding the outline

"To save time and to avoid useless work you must develop the ability to scan, or skim, articles and chapters looking for words, phrases, or sentences that tell you that you have found information pertinent to your topic. If you read slowly and plod through every item in your working bibliography, you will lose much time.

Having a good preliminary outline will aid you greatly in scanning. (...) You will pass your eyes very rapidly down a column of print, not absorbing what you read but looking for a clue that alerts you to information useful to your paper. When you *encounter such a clue, stop scanning and start reading carefully in order to take notes."*

[Willis 1977, pp 49-50]

To assist the end user, we require, therefore, a software system which exploits the outline of the intended essay, integrated with instantaneous navigation within and across relevant areas of interest. A user should at least be able to:

Extracting:

- Get help from functions that zoom in on material within documents more relevant to the subject being developed. (This would be a substitute for skimming).
- Create outlines that capture the structure of input documents as the student unveils it.
- Easily create and extend an outline which will eventually become the final product.

Marking:

- Emulate the function of highlighting by capturing segments of text.
- Generate bookmarks and easily integrate them into existing outlines.
- Be able to easily go back to the exact position in the text where the notes, segments and bookmarks were created.

Annotating:

- Write notes tied to specific positions in the text.
- Organize the notes, segments and bookmarks within the hierarchical section of the outline.

Figure 1.4 below illustrates these requirements from a user point of view.

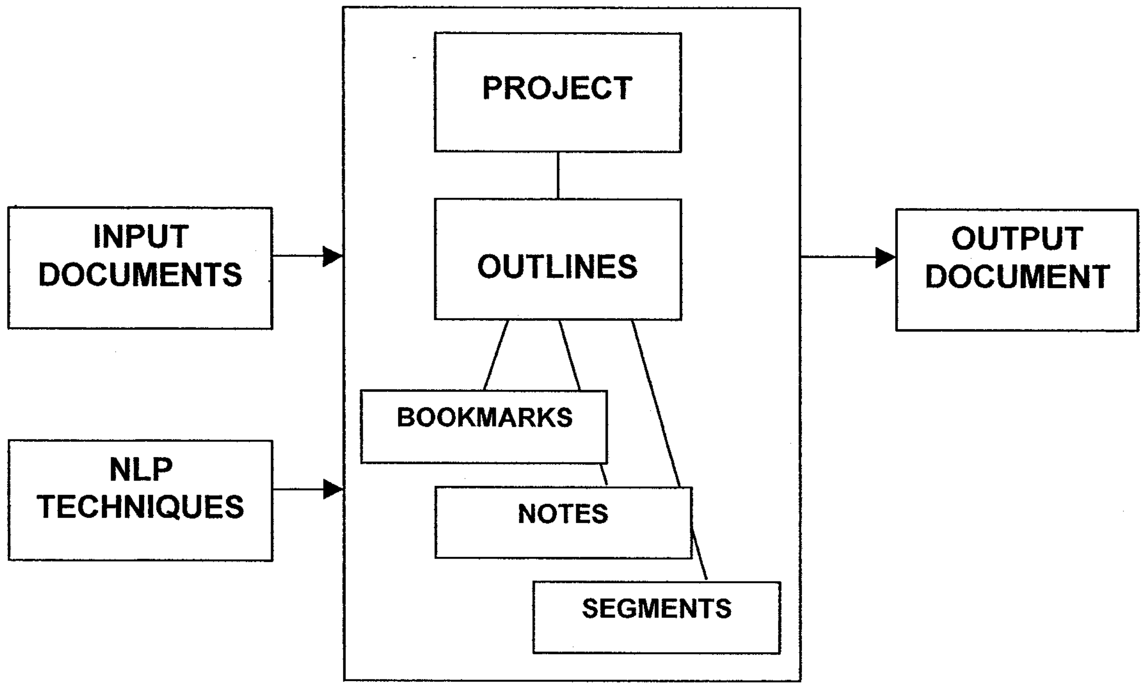

Figure 1.4 – Preliminary Model

The terms used in Figure 1.4 are defined in Table 1 for the reader's convenience.

Project	Corresponds to a unit of work like an essay or paper that will culminate in a final and complete document, including introduction, conclusion and Bibliography.
Outline	A representation of the structure of an input document such as a book or of an output document being worked on. An outline contains bookmarks, notes and quotations (segments).
Bookmark	Like its physical counterpart, a way for the user to mark a specific position within a document and to return to it at will.
Note	Text created by the user. It mimics yellow stickers or handwritten notes in a book.
Segment	A fragment of text from a document that mimics highlighted text or could eventually serve as a quotation.

Table 1 - Preliminary terms definition

Armed with our shopping list, we will, in the next chapters, review what has been done so far to meet these objectives in academic research and in the industry. We will then propose and explain our own solution and show how far NLP techniques can be used to meet these objectives. Our solution will comprise a prototype applying an architecture exploiting the outline paradigm to support adequate NLP techniques and the use of some of these techniques. Field experimentation of this prototype and its results will be described. We will end with a description of the new horizons that we have uncovered and of the future work that we propose.

1.5. Organisation of the thesis

This thesis is organized as follows: Chapter 1 gave an overview of the problem and the proposed solution. In chapter 2, we review previous work in related areas. Chapter 3 and 4 present our solution in detail and finally chapter 5 presents the evaluation of the system. A summary of this thesis can also be found in

Advances in Artificial Intelligence, Proceedings of the 19th Conference of the Canadian Society for Computational Studies of Intelligence, Quebec City, Quebec, Canada, June 2006 (Springer-Verlag, Berlin, 2006)

Chapter 2

2. Literature and Industrial Products review

Our review will bear on two domains, assistive reading and writing tools for the visually handicapped and document exploration tools based on Natural Language Processing. Assistive tools will be reviewed with the requirements laid out at the end of Chapter 1. Document exploration tools such as automatic summarization will be reviewed with the specific perspective of how far it can be used to help a student do a first review of texts and documents and reduce the time he/she takes to do so.

2.1. Assistive tools for reading and writing

The most appropriate source for review is the proceedings of the annual International Technology and Persons with Disabilities Conference of the Northridge Center on Disabilities from the State University of California[8]. This conference, commonly known as **CSUN** (California State University Northridge), is the most renowned and comprehensive conference on the topic. It is currently organizing its 21st conference to be held in Los Angeles in March 2006. Reviewing what was presented at CSUN is a very good indication of what is going on in research and industry. Our review spanned from 1998 to 2005 and we searched for prototypes, software applications and articles, anything bearing one way or the other on the subject of our work and related to the visual handicap.

As would be expected from the importance of technology for the visually impaired, the topic of hardware and software applications was the one that was given the most attention. Before reviewing this topic in detail, two other topics deserve mention. First, it is becoming more and more important to have books and documents in an electronic form. This raises the question of electronic textbooks, their accessibility, how they are produced and other important issues like copyright laws. But the topic that has been given by far the most attention over the recent years has been digital talking books and the **D**igital **A**ccessible **I**nformation **SY**stem[9] (DAISY) standard. This standard allows for the encoding of audio and text through XML mark-up tags in a document so as to produce accessible and navigable multimedia documents. The net results are Digital Talking Books, digital textbooks, or a combination of synchronised audio and textbooks.

In general, the encountered software applications were either geared to reading or writing, but not both. None incorporated NLP techniques to assist users in reading and writing tasks, except textHELP[10] and WYNN[11] and they were restricted to homonym checking and word prediction. The following systems were the most interesting: WYNN[12], Draft Builder[13], textHELP and Kurzweil[14] 1000. A description of each follows.

---

[8] http://www.csun.edu/cod/conf/index.htm
[9] The following link provides a Good introduction to DAISY
[10] http://www.texthelp.com/
[11] http://www.freedomscientific.com/LSG/products/wynn.asp
[12] http://www.freedomscientific.com/LSG/products/wynn.asp
[13] http://www.synapseadaptive.com/donjohnston/draft_builder.htm This product is presently discontinued.
[14] http://www.kurzweiledu.com/

We will restrict ourselves to features pertinent to our thesis. At the end of this section a comparison table will summarize their features.

2.1.1. WYNN

WYNN is a tool to read and create documents mainly targeted for Learning Disabled students. Figure 2.1 shows screenshots of the three main screens of WYNN. It uses bookmarks, notes, highlights and outlines. It can create bookmarks. However, they are single level and the bookmark can be of one word only. The function 'Create Outline' opens a blank window where the user can create a standard point form outline.

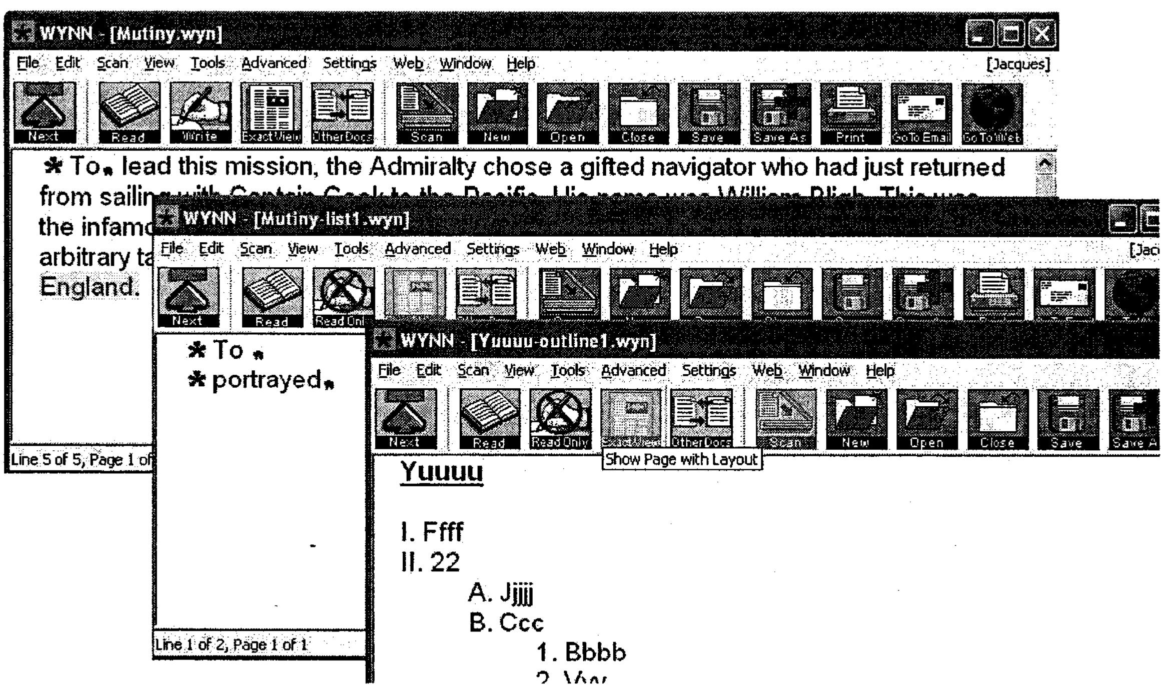

Figure 2.1 – Sample screens of WYNN

A listing of the bookmarks or highlights will create auxiliary documents that the user can review and print.

The main problem of WYNN is that there is no facility to go to the position of the bookmark in the original document. There are no pointers and there is no way to extract multi level bookmarks into a true outline nor can the user construct an outline from several documents grouped into a project. The user can search for bookmarks or highlights but has to cycle through them like when the user searches for a word and goes to each successive occurrence.

WYNN also includes word prediction. This technique allows the application to predict the most likely word to be typed given the previous context. The user types a letter and the program offers a list of the most likely words beginning with that letter. If the required word is on the list, it can be selected. If the word is not on the list, typing the next letter will bring up a different choice and so on. It is not clear if a language model of the user is actually built, or if a simple dictionary look-up is performed.

2.1.2. eClipseReader

eClipseReader[15] is a sophisticated DAISY Book reader. Figure 2.2 gives a feel for the user interface. This software enables print disabled persons to play talking books with the possibility to navigate within chapters, pages, headings and phrases. Bookmarks are available but not as part of an outline. They are called 'Markers' and they have to be identified by numbers or user defined names. As its name suggests, this software makes no provision for assisting the user in writing tasks. eClipseWriter, also available from the same company, is not intended for that purpose. The latter is a simple authoring tool for sighted persons. It allows the creation of literary products following the DAISY format and standard.

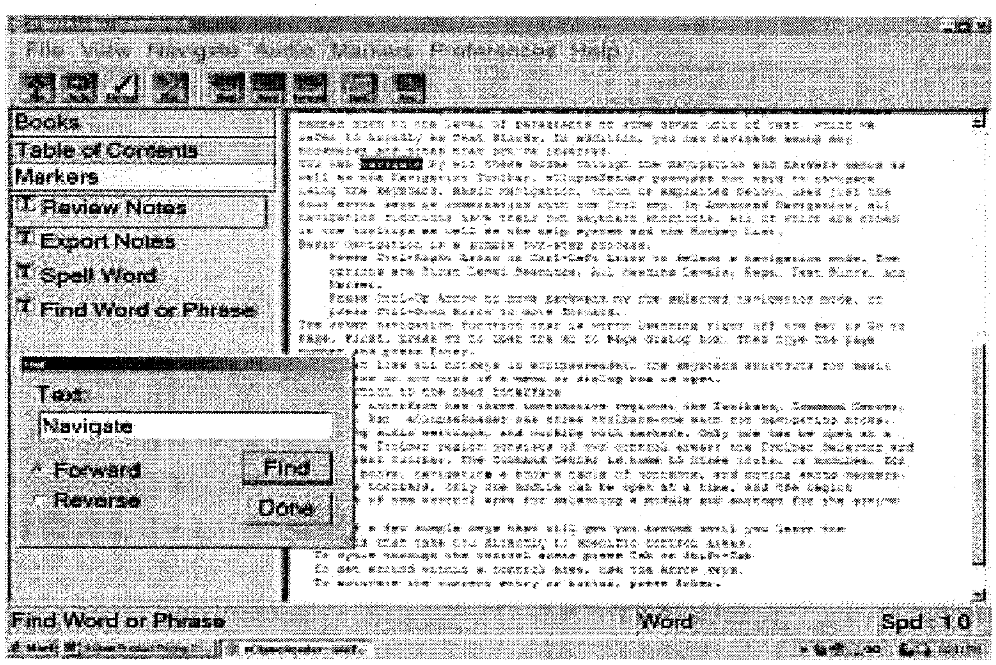

Figure 2.2 – eClipseReader screenshot

[15] http://www.eclipsereader.com/

2.1.3. Victor Reader Soft

Victor Reader Soft[16] is also a DAISY reader like eClipseReader in Section 2.2 but an elementary one with an awkward interface. There is no possibility of creating outlines, extracting bookmarks to outlines or saving the bookmarks, at least in version 1.5.8. Bookmarks are numbered. The user must remember which bookmark corresponds to which number and provide that number to go to the text corresponding to the bookmark. Figure 2.3 shows the only screen available in the application.

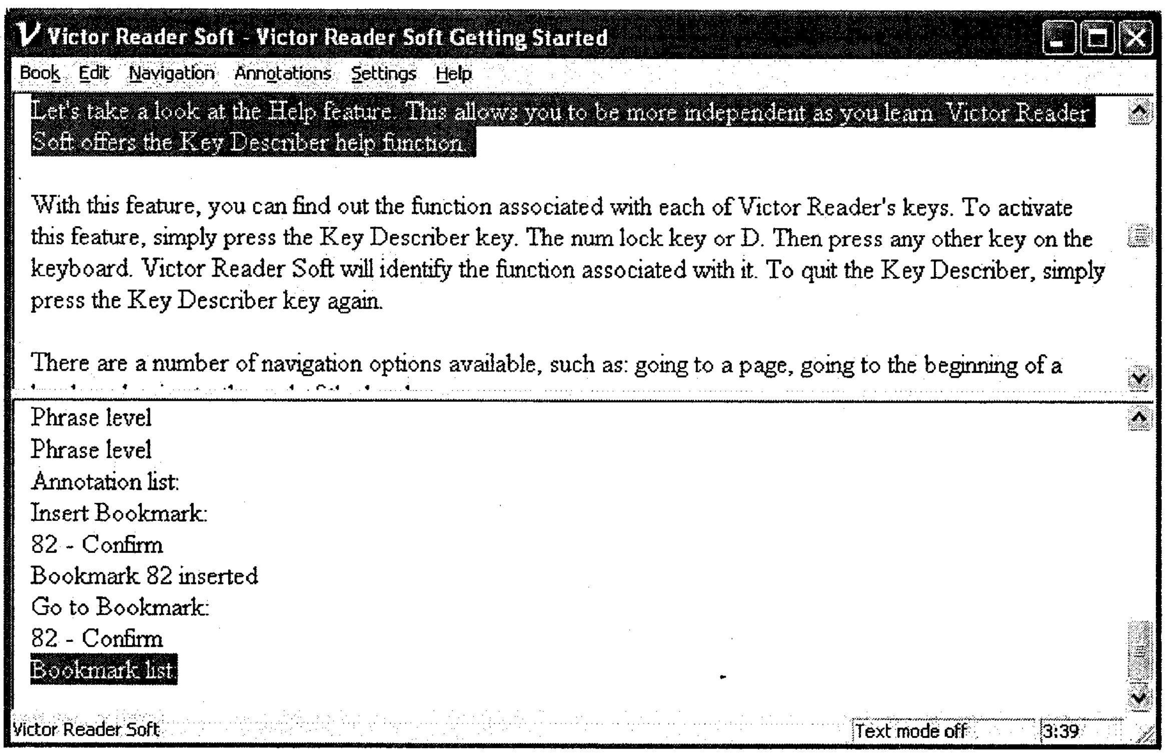

Figure 2.3 – Victor Reader screenshot

2.1.4. Draft:Builder

Draft:Builder is an outlining application that targets beginners, those learning to write (Primary and early Secondary School level).

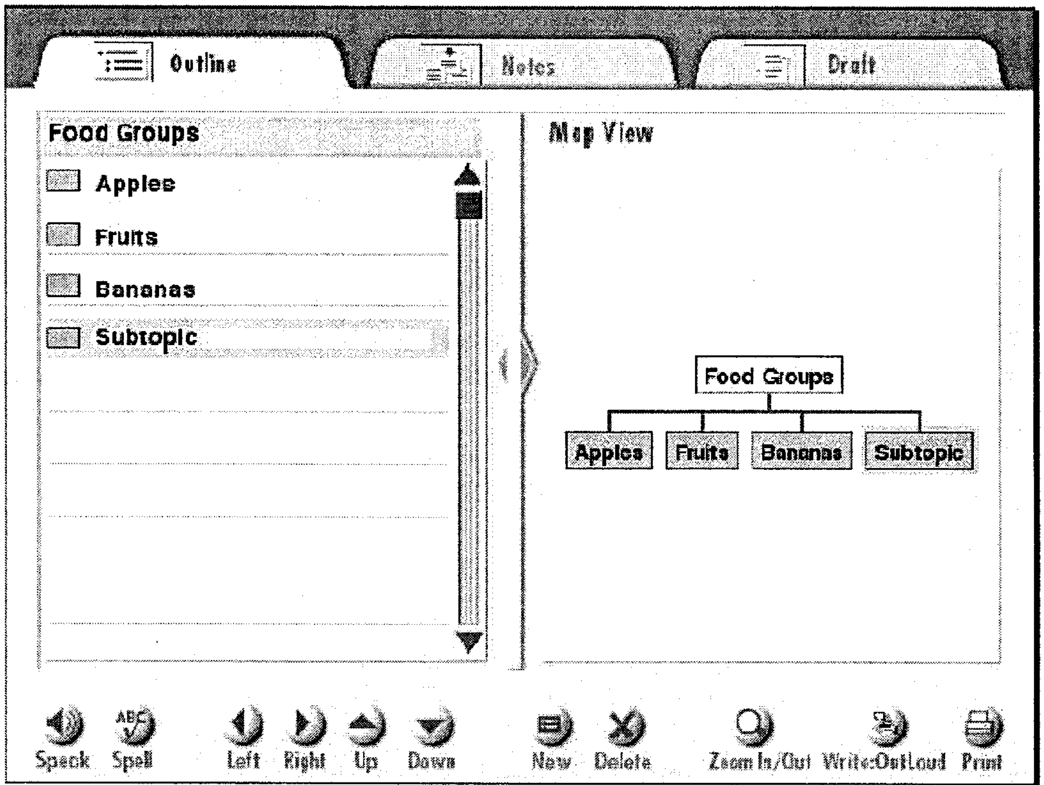

Figure 2.4 – Draft:Builder screenshot

It is very appropriate for young sighted Learning Disabled students. The user can build an outline within two alternate views as can be seen from Figure 2.4. He can also write notes that can be assigned to the elements of the outline. However, documents and pointers to the contents of documents do not exist. Essentially Draft:Builder is a writing aid with no integrated reading facilities and a very visual interface requiring extensive use of the mouse (drag and drop).

[16] http://www.humanware.ca/HW/en/victorsoft.asp

2.1.5. Kurzweil

Kurzweil offers two products that are very similar in functionality but target two different category of users. Kurzweil 1000 is reading software for the blind while Kurzweil 3000 is specifically targeted to users with learning difficulties like dyslexia, attention deficit disorder (ADD) and other literacy difficulties. The latter has a very visual interface relying heavily on color. The bookmarking and outlining facilities are sensibly the same as in Kurzweil 1000. For this reason, we will only consider Kurzweil 1000 here. We will do so in greater detail since this is the software that comes the closest to what we want to achieve.

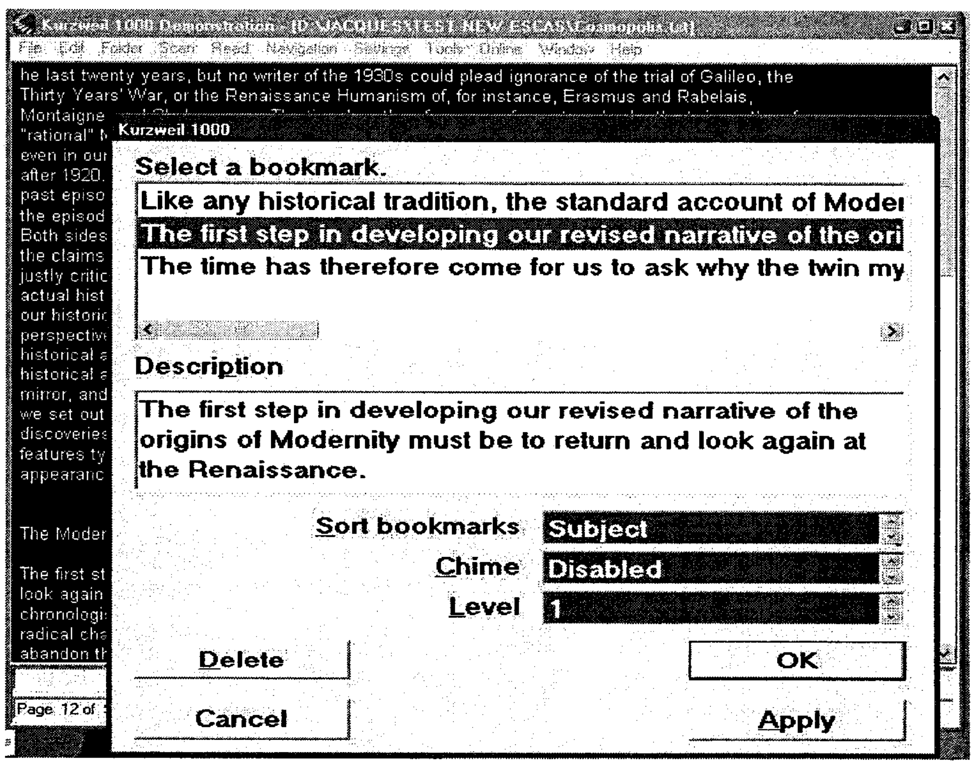

Figure 2.5 – Bookmark navigation screen in Kurzweil 1000

The main purpose of this software is to offer elaborate facilities to convert documents to text and to speech with scanning functions, Optical Character Recognition and synthetic text-to-speech software. A scanner is mandatory when the software is installed. Writing and editing tools provide an audible text editor.

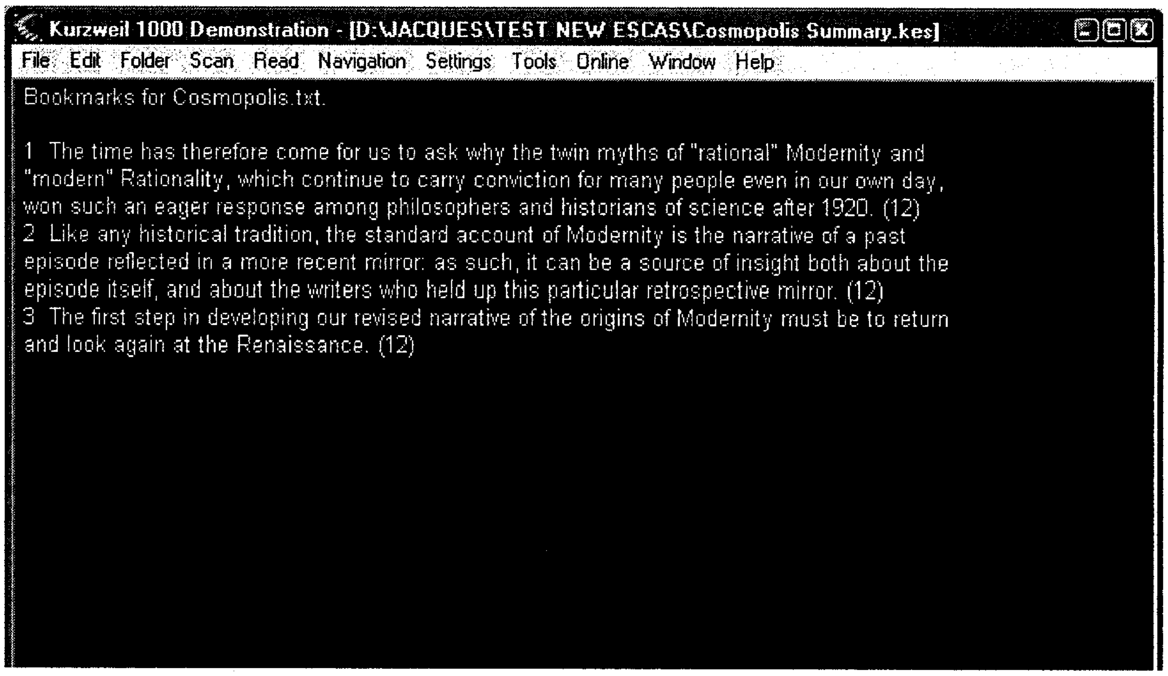

Figure 2.6 – Bookmarks extracted to an auxiliary document in Kurzweil 1000

While reading, the user can create bookmarks from the menu or by pressing Ctrl B. These bookmarks can be sorted, used for navigation and structured into multi-level bookmarks by changing their levels (Figure 2.5). The multi-level bookmarks and notes can be extracted to produce an outline in another document (Figure 2.6). However, Kurzweil 1000 is not an outliner. The user cannot refer back from the outline to the document where the bookmark was created nor can he construct an outline from several documents grouped into a project. The creators of Kurzweil specifically refer to their 'Study Tools' as tools for 'reading reinforcement'.

2.1.6. textHELP

textHELP develops software products that are designed to assist individuals to improve their reading and writing abilities. They are designed as 'floating toolbars'. The core of these systems is speech output. They should be classified as dyslexia software or text to speech software. There are no bookmarks or outlining facilities. These products include Read and Write (Figure 2.7)[17], a utility program whose features include speech output, spelling and homonym checking, word prediction, and thesaurus capabilities, and WordSmith, a scanning program that interfaces directly with Microsoft Word. Homonym support provides auditory and visual reinforcement of commonly confused like-sounding words. To eliminate the confusion of homophones, the program color codes confusable words and lists possible alternatives with audible definitions and sample sentences.

Read & Write 7.1

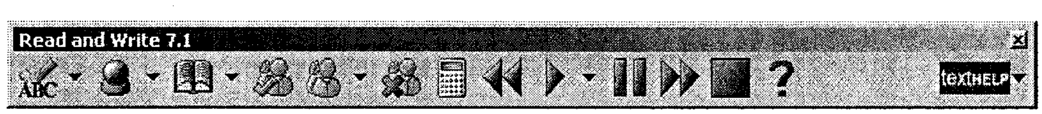

Figure 2.7 – textHelp floating Read & Write 7.1 toolbar

[17] http://www.texthelp.com/rw7.asp?q1=products&q2=rw7

2.1.7. Summary

Table 2 below summarizes the characteristics of the applications described above that are currently available to the visually handicapped with respect to the criteria we established in Chapter 1. We did find software that succeeds very well in providing valuable assistance to readers. However, our purpose here is to assist a user in manipulating several documents in an integrated fashion within a single project, capitalizing on notes and bookmarks that he has already created while reading, exploiting the ability to refer back instantly to documents so as to be able to explore the context, in short, having a rich outlining environment seamlessly integrated with his reading environment. To our knowledge, no currently available software is able to meet our criteria.

Criteria:	**WYNN**	**Kurzweil 1000**	**textHELP**	**eClipse**	**VR Soft**
Read Documents	●	●	●	●	●
Generate and save bookmarks.	●	●		●	
Easily create outlines that capture the structure of input docs as the student unveils it.	●	●			
Emulate the function of highlighting by capturing segments of text.	●	●		●	
Write notes tied to specific positions in the text.		●		●	
Easily and instantaneously be able to go back to the exact position in the text where the notes, segments and bookmarks were created.		●		●	
Easily integrate bookmarks into existing outlines.					
Organize the notes, segments and bookmarks within the hierarchical section of the outline.		●			
Write Documents	●	●	●		
Easily create and build on an outline which will eventually become the final product or be heavily used to create it.					
Get help from functions (using NLP) that zoom in on material within input docs more relevant to the subject being developed. (This would be a substitute for skimming)	Word Predic- tion		Word Prediction, Homonym checking		
Bookmark the relevant material thus obtained.					

Table 2 - Software features comparison

2.2. Document exploration tools

2.2.1. IR/Paragraph Retrieval

NLP techniques seem to be non-existent in the products we reviewed in section 2.1 with the exception of word prediction and homonym checking.

2.2.2. Automatic summarization

> *"The goal of automatic summarization is to take an information source, extract content from it, and present the most important content to the user in a condensed form and in a manner sensitive to the user's or application's needs."*
>
> [Mani, 2000]

In our case, our "*user's or application's needs*" is to save time in extracting information from sources to write essays where time is a critical issue. The business of professional abstractors is to quickly capture basic information in documents. It would be interesting here to compare the user tasks outlined in chapter 1 with the way abstractors work. According to [Mani, 2000], professional abstractors:

- Use a top-down strategy. (They never attempt to read a document from start to finish.)
- Use the structural organization of the document and discourse-level rhetorical relations to link relevant text elements.
- Exploit shallow features such as cue phrases, location cues, title cues and in-text summaries.
- Take notes and underline relevant text material.
- Interleave writing abstract with reading document.
- Cut and paste using standard sentence patterns.
- Use both syntactic and semantic level representations.

There is an obvious similarity and complementarities between the two approaches which reinforces the fact that automatic summarization functions would be of great help. Many of the aspects of how human abstractors work is used in automatic summarization systems. Shallow features, discourse-level representation, cut and paste are examples. However, they are used singly, they are not integrated.

Our objective in reviewing automatic summarization tools is to first try and find components or applications that can be used as is or incorporated in our proposed application. They need to run on the Microsoft Windows platform. The following extract based software has been identified:

- Automatic Summarizer in MS Word
- Copernic (uses Extractor)

- InXight
- NetOWL

2.2.2.1. Automatic summarizer in MS Word

This function, only accessible from within Microsoft Word, is the most commonly used baseline against which the research community compares their work [Marcu, 2000], [Saggion, Lapalme, 2002]. This summarizer is only effective with very homogeneous and well structured texts. Most systems or prototypes perform better. Due to its inaccessibility for integration into another application and its poor performance as a summarizer, it needs to be discarded as an option.

2.2.2.2. Copernic and Extractor

Copernic[18] and Extractor[19] [Turney, 1999] are considered together because Extractor provides a DLL (Dynamically Linked Library) with an API[20] (Application Program Interface) and is available for commercial software companies, for embedding in commercial software products and Copernic is a good example of such an application.

Extractor is software for automatically summarizing text, developed by the Interactive Information Group of The National Research Council of Canada. Extractor takes a text

---

[18] http://www.copernic.com/en/products/summarizer/index.html
[19] http://iit-iti.nrc-cnrc.gc.ca/collaborations/licensing-licences_e.html

file as input and generates a list of key words and a list of key sentences as output. Extractor extracts from 3 to 30 key phrases from a document written in English, French, German, Japanese, Spanish, or Korean. Since this software provides a DLL with an API that has been used with success in other applications like Copernic, it should definitely be retained for further evaluation.

2.2.2.3. InXight

The InXight summarization module is part of a more complete environment comprising other modules[21]. InXight provides Software Development Kits (SDK's)[22] for six components of which three are directly related to Natural Language Processing. LinguistX provides multilingual text analysis and information retrieval through tokenization, stemming, part of speech tagging and noun phrase extraction. ThingFinder is a text-analysis application that automatically identifies tags and indexes named entities in a document-such as persons, places, addresses and dates. The third component, Summarizer[23], has the following characteristics:

1. Establishes key phrases embodying the document concepts.
2. Allows the user to control the weight of these phrases according to their interests.
3. Establishes criteria for selecting sentences using the conceptual content (key phrases) of the document, the structure of specific document types, the number of thematic

---

[20] Application Programming Interfaces make it easy for software developers to tightly integrate extraneous components into their own applications.
[21] http://www.inxight.com/products/
[22] http://www.inxight.com/products/sdks/ SDK's include Application Programming Interfaces (API's) that make it easy for software developers to tightly integrate extraneous components into their own applications.

words and proper names, its location in the document, and the length of the document.

4. Selects key sentences using these criteria.
5. Displays the output based on the length of the sentences or the number of key phrases.

Like Extractor, this product should be retained for further evaluation since it also provides an API for the Windows platform.

2.2.2.4. NetOWL

SRA's[24] NetOwl[25] summarizer offers summarization with the following features:

- Theme-Based Summarization that identifies the main topics of each document and produces a concise version that accurately reflects them.
- Query-Based summarization focusing on the user's specific information needs.
- Adjustable summary length.
- "Trainable" system that adapts to new subject areas, content and sources.

C and Java API's are available that run on Solaris 2.7 or higher and Windows XP

This product would be the third, with Extractor and inXight, to be retained for an eventual evaluation of the API.

[23] The Summarizer SDK is supported on Microsoft Windows XP, Sun Solaris 7 and Redhat Linux 6.2
[24] http://www.sra.com/
[25] http://www.netowl.com/products/summarizer.html

Chapter 3

3. A proposed solution

One practical way to describe a software solution is with a working prototype. Since the purpose here is a proof of concept, we will not address all the features required for a marketable product nor shall we implement all the possible NLP techniques. Doing so would take us beyond the depth and scope of this thesis. We will structure the description of our solution into two parts; the basic architecture and the NLP add-ons. The basic architecture will provide for enhanced navigation, centered on the outline and bookmark paradigms and it can be fully implemented as a first level solution. NLP techniques are expected to convey additional information about a document through pointers to relevant information. This architecture will provide a framework within which such pointers can be structured and used efficiently.

3.1. Basic architecture

As previously discussed in sections 1.4.2 and 1.4.3, since the purpose of the exercise is to produce a structured document, the basic architecture is centered on outlines grouped into "one project". An outline consists of items and sub items organized into the form of a tree data structure. Outlines are discussed in more detail below in section 3.1.2.

There can be several outlines and one of these outlines will correspond to the final document evolved. We shall call it the 'Working Outline'. There is normally one per project. The other outlines correspond to the input documents and are thus called 'Reading Outlines'. When the user reads a document, he will want to capture its structure as he discovers it. He will want to bookmark some sections, make notes and identify parts of the text that could be used for quotes in the final document. This is the purpose of the 'Reading Outline' which cannot exist without an associated document. This document is expected to be in 'Read only' mode. There should be as many Reading outlines as there are documents to be studied for a specific project.

3.1.1. Project

A project is an essay or research paper to be written that will culminate in a final and "complete document for submission". When the prototype is started, the first screen that appears is the main project screen (see Figure 3.1). From that screen the user can either create a new project or open an existing one. One outline is the minimum requirement for a project to exist. When a new project is created, this working outline (initially empty) is automatically created with the same name as the project. Creating or opening a project brings up the only other screen of the prototype (Figure 3.2).

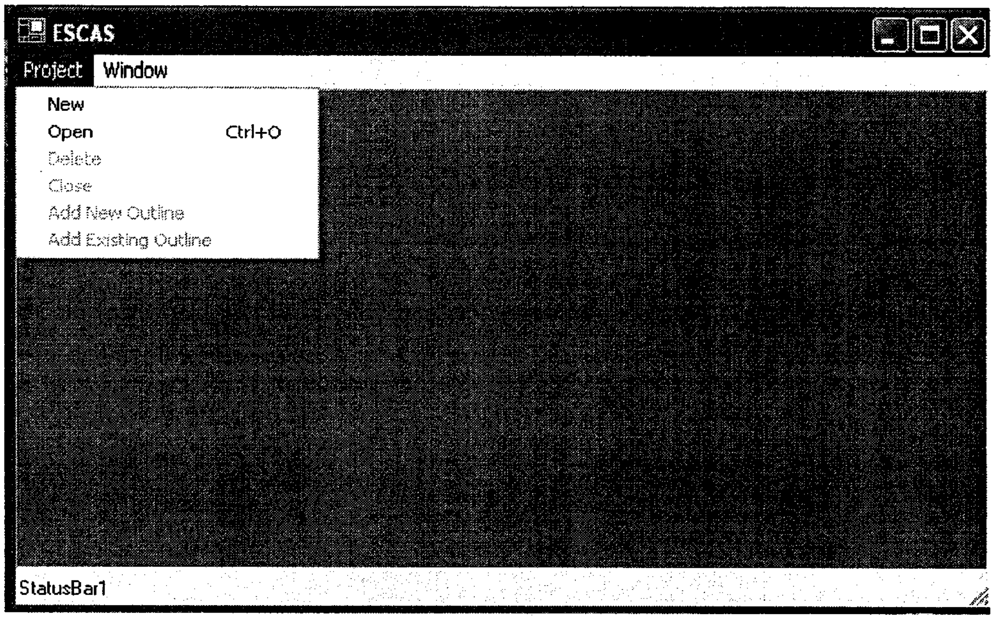

Figure 3.1 – Project Screen of the prototype

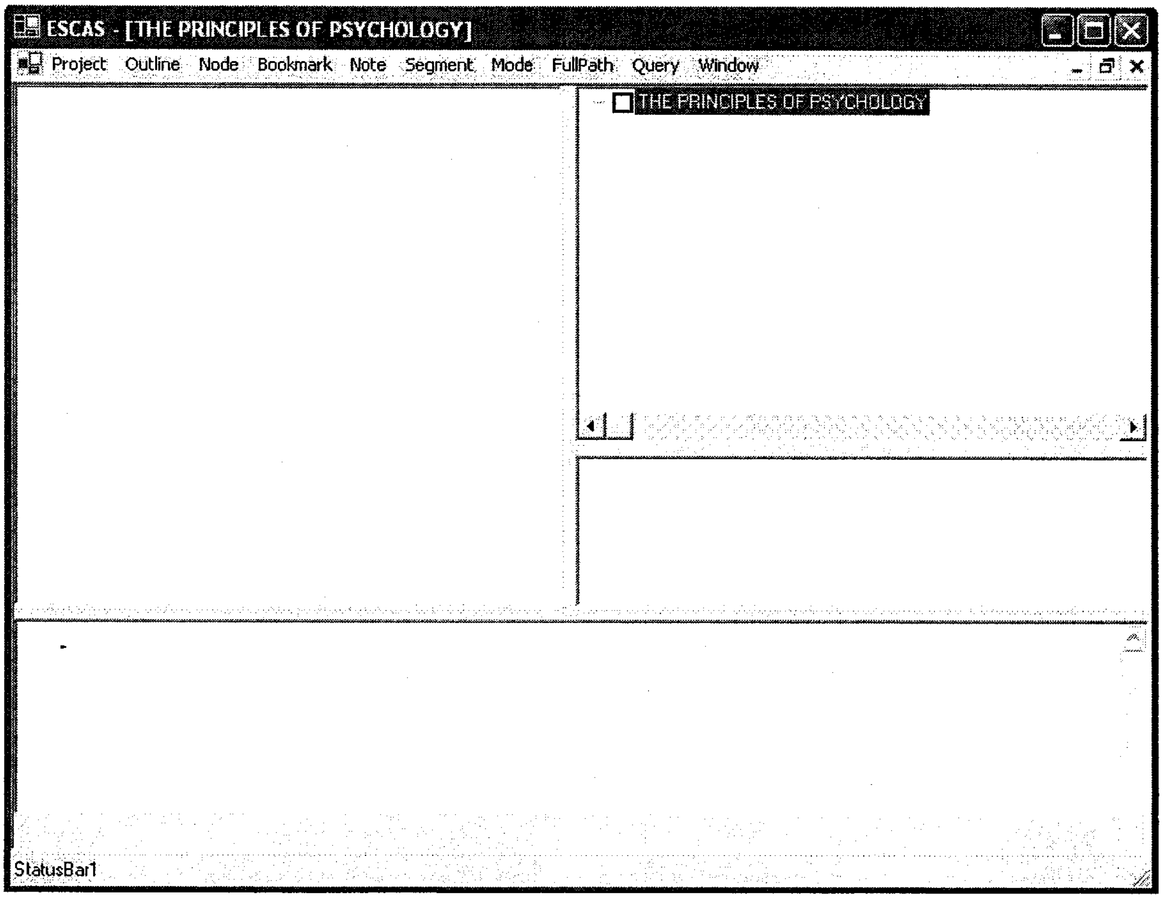

Figure 3.2 – New project with initial empty outline

The initial working outline acts like the Table of contents of the final document. It is expected to grow and be constantly edited until it is ready to be converted to a draft (see Figure 3.3).

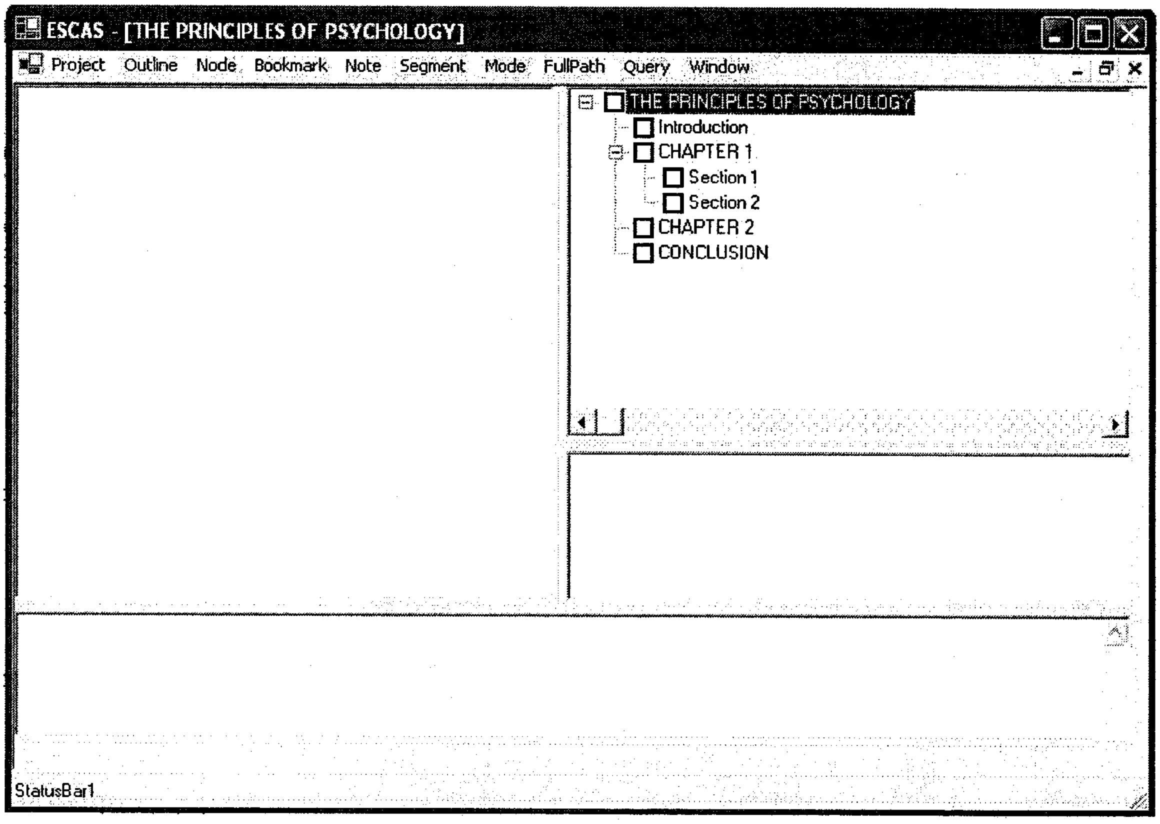

Figure 3.3 – Working outline

Figure 3.3 also gives an idea of the User Interface elements in the main screen. The screen essentially consists of two text areas, a list box and a tree object. The text area on the top left side is for the document being read. The tree on the top right displays the structure of an input document such as a book (a reading outline), or of an output document being worked on (a working outline). The second text area under the tree and on the right is for notes and quotations. The list box at the bottom of the screen is a working area for NLP techniques. When a search is required in either the document area or an outline, the user can bring up a textbox with CTRL F to enter the search string. Pressing the F3 key repeats the search from the last occurrence to the end of the document.

3.1.2. Outline

An outline consists of items and sub items, also called nodes, whose structure can be compared to a tree. The nodes can be bookmarks, segments, notes and user text. Figure 3.4 is an example of a Reading Outline linked to Chapter 9 of the 'Principles of Psychology' of William James and containing the different types of nodes.

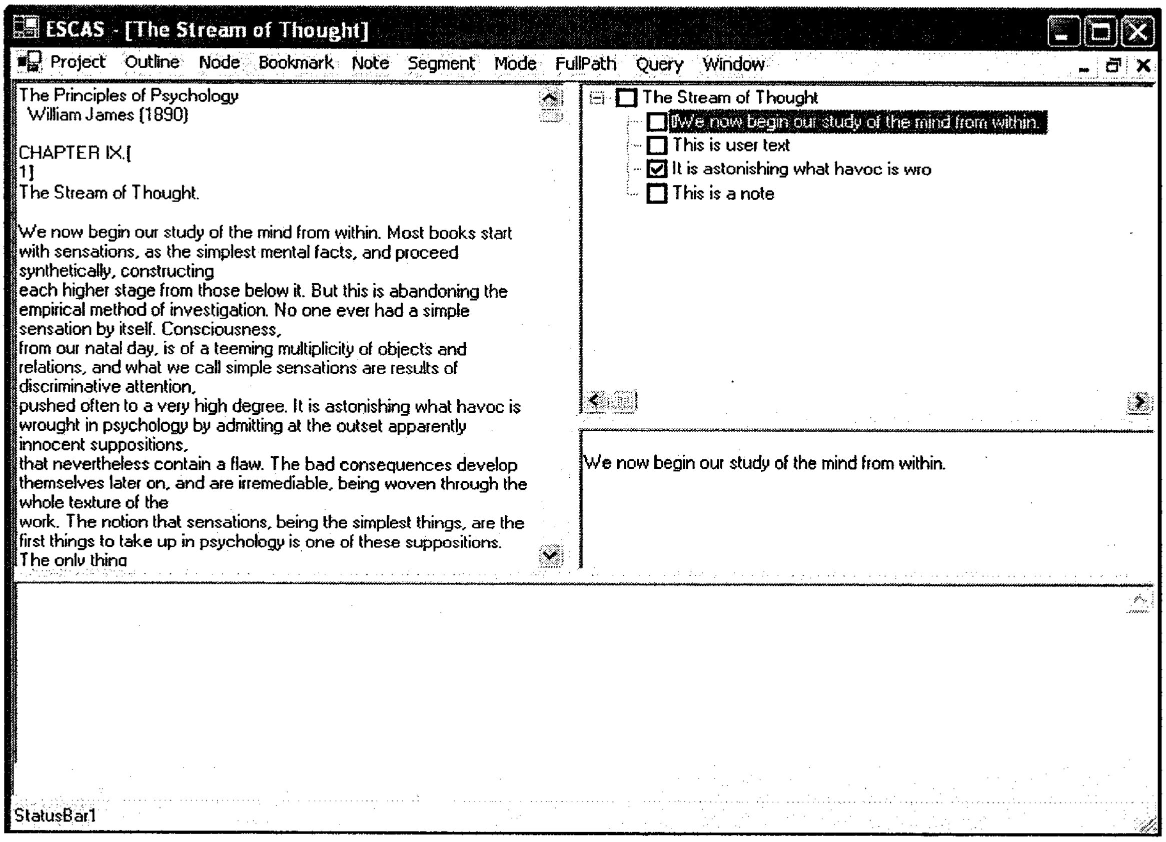

Figure 3.4 – Reading Outline

We will continuously refer to this example in the following paragraphs when explaining these concepts. These nodes or sub-items can all be copied, promoted to a higher level, demoted to be the sub item of another item or deleted.

Simple standard search functions are available for quick positioning in the outline. The multilevel outline can thus be restructured so as to give full control to the user in the creation process. At any time, an outline can be exported into an HTML file with links corresponding to the main and sub headings. This HTML file is particularly useful for collaborative work with other students. Preliminary notes as well as advanced work can be shared. The outline can also be exported to a text file that can be used as a starting point for a final document.

3.1.3. Bookmark

A bookmark is a place holder, a specific position within a document that a reader can return to. Students with no visual handicap use several artefacts to generate bookmarks like 'marking the book' with a highlighter. When they need to go back to review these places, they can skim through the book.

In our solution, when the user wants to ensure he can subsequently return to where he is in the text, he can generate a bookmark through the menu or by using Alt B. This bookmark will be assigned to the current node in the current outline, in the form of a string corresponding to a number of characters starting from the position in the text where the user was positioned. At any time, pressing Enter when the Bookmark is selected will return the user to the same position in the text.

In Figure 3.3, 'It is astonishing what havoc is wro' is a bookmark. The number of characters in the bookmark is arbitrarily set to 35. User options could be created where the user could select a different bookmark length or choose to capture the text from the cursor to the end of the sentence or the text of the complete sentence in which the cursor resides. The tick next to the bookmark allows the screen reader[26] to give feedback on the

[26] A screen reader is software that verbalizes through synthetic speech text appearing on the screen.

type of node to the user. Another feedback mechanism on node type is one or two beeps when the node gets the focus. The user has the option to assign a particular feedback mechanism to a type of node of his choice or turn off the mechanism altogether.

3.1.3.1. Note

The user can write notes tied to a specific position in the text. Pressing Enter on a text node will return the user to that position. Alt N or the 'Note' menu option will take the user to the textbox situated under the outline box and clear it for the user to compose his note. Pressing F2 will save his note and assign it to the current node in the current outline with the note as the identifying string. This allows the mimicking of yellow "stick-notes" or handwritten notes in a book.

3.1.3.2. Segment

A segment is a fragment of text marked on a document that mimics highlighted text. It is usually very difficult for visually handicapped users to control the selection of text with the shift key, let alone the mouse. Here, he can mark text at the start of a fragment that he wants to capture by pressing an opening key (Function Key 1). He can then move around without fear of losing his selection. He then goes to the end of the fragment and presses a closing key (Function Key 2). This will create a segment that will be assigned to the current node in the current outline. This time the string showing in the entry will correspond to the text starting and ending where the keys were pressed respectively. A segment allows the user to return to the position where the segment was extracted and explore once again the context of the segment if required. In Figure 3.3, 'We now begin our study of the mind from within.' is a segment marked on the text. Since a segment can be of any length, the Note text box is used to display it.

3.1.4. User text

Some items or sub items like the different levels of headings and subheadings in a table of contents need not necessarily be tied to a position in some text. Typical ones are 'Introduction' or 'Conclusion'. These items are qualified as user text in opposition to notes that are also bookmarks and they can be of any length. They are regular nodes that can occur at any level. When an outline is created, at least one 'user text' type root node is automatically created with the name of the outline.

3.1.5. Building a draft

Since notes, segments, user text and bookmarks can be moved up and down, copied to other nodes or to other outlines, it is very easy to organize them within the hierarchy of an outline. This is especially useful when dealing with a working outline. This outline corresponds to the final document being created. The user can thus build a skeleton and flesh it out with notes and quotations (segments) in an iterative fashion. He can copy nodes from reading outlines just as he would select and reorder note cards.

At all times, the user can create a document in the format of his choice to preview what the final document might look like. In Figure 3.5, the list box on the top left of the screen contains an output which corresponds to the outline on the right. Adding user driven formatting would be a necessary step. Should he make changes to that output preview document, however, they would not be reflected in the original source outline. For the outline and the document to reflect the same information at all times such that the facilities of the outliner can be used, the user has to make changes in the outline and regenerate the document from the outline rather than make changes to the document, which changes will be lost if the preview is regenerated from the outline.

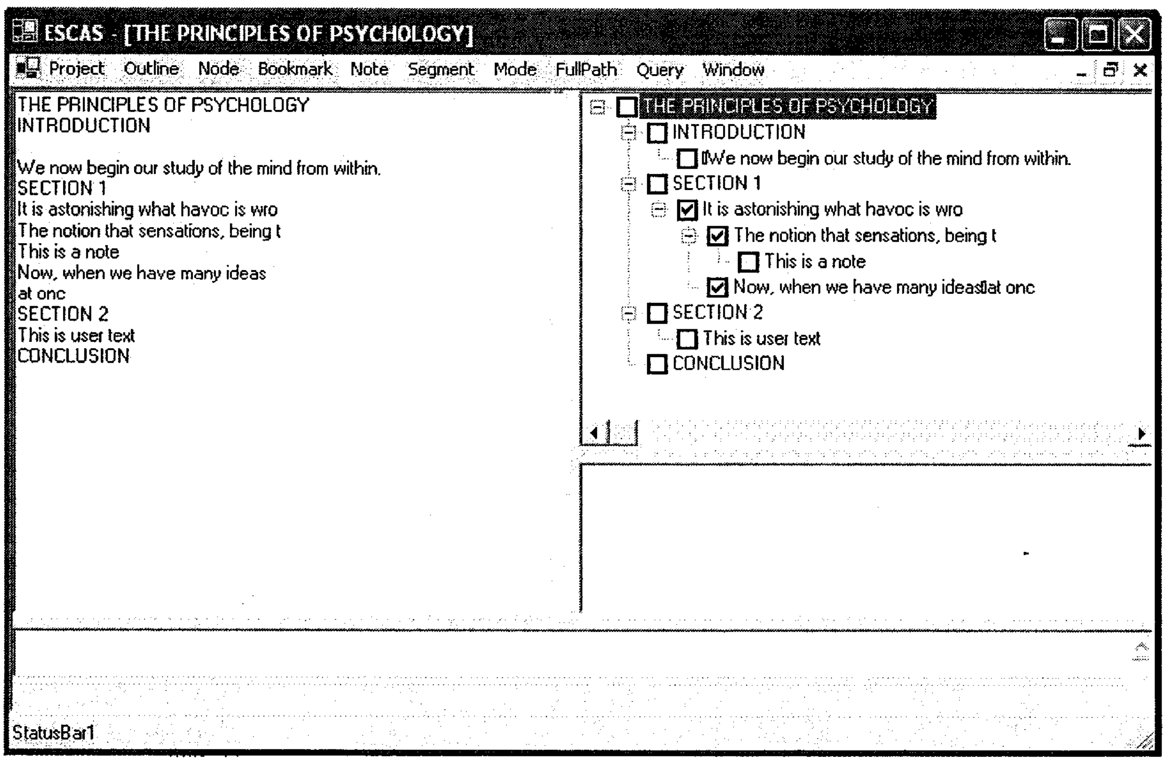

Figure 3.5 – A draft document

Figure 3.5 is an example of how the working outline in Figure 3.3 was enhanced with nodes from the outline in Figure 3.4 and reorganized.

3.2. NLP techniques

Natural language processing can contribute in many ways to Document Understanding. Our aim is to choose the right NLP techniques to zoom in on material within an input document that is particularly relevant to the subject being developed by the user. In the case of sighted users, this can be achieved by skimming or fast reading through the material. Our intent here is not only to replace skimming with what NLP can offer but also go beyond skimming and implement suitable functionality that could be of help even to those who have no handicaps.

Obviously, these techniques will have to be selected and evaluated in terms of their usability and applicability in real life and real applications. But more important, they have to be amenable to the basic architecture we have chosen, an architecture that tries to help the user exploit fully all the 'intelligence' that these techniques bring with them. It is important to stress here the respective roles that we want to assign to the "machine" as opposed to "humans":

> *"...it is very likely that machines and humans may have overlapping but not identical capabilities.*

For example, computers are somewhat better than humans in sifting through large quantities of data, whereas humans are much better than machines in making inferences based on context and world knowledge."

[Mani, 2000]

More practically, in accordance with the experience of the RALI[27] Laboratory who reoriented their Machine Translation applications to assisting professional translators rather than substituting themselves to the latter, the user will have to stay in control of the application.

3.2.1. Automatic Summarization

The first technique we considered is automatic summarization. Commercial software currently available is all based on the extraction of right sentences to be copied over to the summary (see section 2.1). However, there are some problems with extraction.

- Dangling anaphora can occur if an extracted sentence refers to another one that is absent in the summary.
- The gaps created by the sentences that have been dropped can result in incoherent text.
- Some elements of texts cannot be arbitrarily divided like tables and itemized lists. If they are, the summary will look absurd.

[27] Elliot Macklovitch, personal communication, Feb. 2002.

In other words, semantic-level aggregation and generalization are typically absent because it is a hard task. While current systems may be useful in some cases, current research in summarization has to progress towards better quality output.

There is another problem, not with summarization per se, but with the way current automatic summarization software present summaries when the latter have to be integrated into our outline architecture. The extracted sentences are combined into one text but there are no pointers provided from the extracted sentences to the source text which would allow the user to return to the source text for contextual user exploration. The extracted sentences do not lend themselves readily to be assigned to outlines such as the bookmarks in our solution. For this reason, we did not include in the present prototype any external summarization component. We rather considered other techniques like indexing and paragraph retrieval which we describe in the next section.

3.2.2. Indexing

When skimming, the user is looking for specific information quickly. An obvious support is the automatic generation of back-of-the-book indexes that provide pointers in a document for words listed in alphabetical order.

3.2.2.1. One word indexing

In a standard index at the end of a book, the page numbers where a specific word can be found are listed. In our solution, we create our own index containing all words besides stop words. Each word (the index entry) is a text node. Under that node, instead of a page number, the user can see the actual text which contains the entry. By just pressing Enter on an occurrence, the user is taken to its exact location in the text. In addition, it can be very handy to scan through a list of sentences rather than to have to keep switching to the text from a list of page numbers.

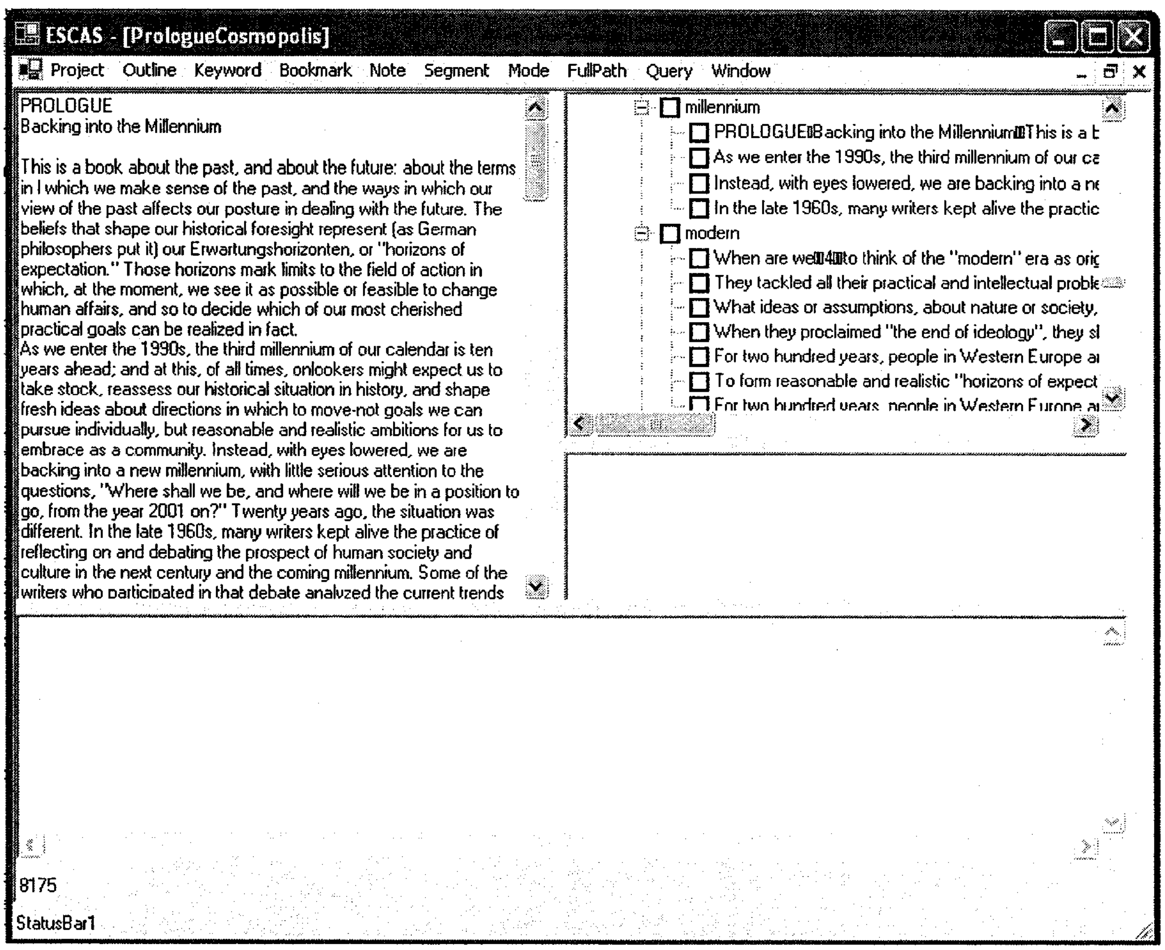

Figure 3.6 – Screenshot showing the use of one word indexing

3.2.2.2. Multi-word indexing

One of the major objectives we set for ourselves was to get help from functions that zoom in on relevant material within input documents so as to provide a substitute for skimming. Most of the time, the user is looking for material that is characterized by a combination of two or more words rather than just one. For example, no one would contest that 'Supreme being' together is more epistemologically specific than 'Supreme' and 'Being' taken separately. The idea here is to establish the set of word combinations that reflect the meaning of the document.

In Natural Language Processing, the study of 'collocations' deals directly with the combination of words.

> *"A collocation is an expression consisting of two or more words that correspond to some conventional way of saying things... Collocations are characterized by limited compositionality. We call a natural language expression compositional if the meaning of the expression can be predicted from the meaning of the parts."*

[Manning, 1999]

Good examples of collocations are 'white wine' or 'kick the bucket'. Collocations are of interest to us for two reasons. First, collocations are important for *'computational* *lexicography (to automatically identify the important collocations to be listed in a dictionary entry)'* [Manning, 1999]. Being a list of words combined together in a significant way, they can be part of an index and thus lend themselves readily to being in the form of bookmarks that a user can easily access. Second, since the process of generating collocations uses a system of ranking which integrates the importance of specific words within a document, this process produces a set of keywords. This set is richer than the one produced by Extractor. In fact, this obviates the problem we encountered with summarization, whereby the user has no pointers to the original document.

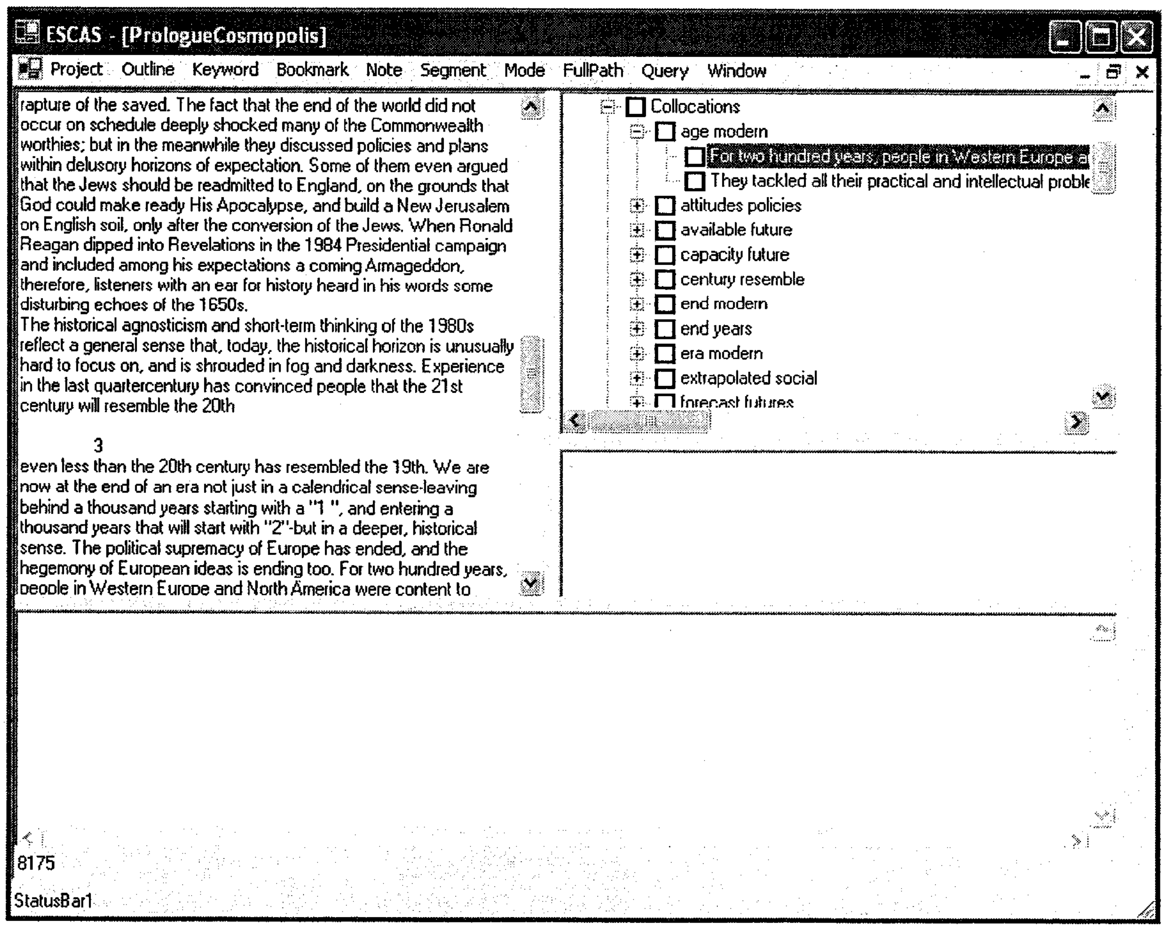

Figure 3.7 – Screenshot of collocation example

Our purpose is not to try and find collocations in the most common sense of the word. Our research is rather to investigate a new way of using collocations, more specifically to see if the collocation algorithms can produce a set of word combinations that, when used as an index, provides additional help to the student when the latter wants to get acquainted as quickly as possible with a document.

3.2.3. Paragraph Retrieval

The idea behind paragraph retrieval is to suggest to the user the places where he would start reading that would be the most relevant to the subject being treated. If he could do a well structured and directed reading of the material, he would save time by having less to read or not having to read the material more than once. The question is how to provide guidance in this regard. This problem can be reformulated into an Information Retrieval (IR) task where the query is the subject of the thesis or paper being written. The document being studied is parsed into a set of paragraphs and the paragraphs most relevant to the query are retrieved. Different statistical techniques of IR can then be used. [Sparck Jones, Willett, 1997]. Practically, the user can provide in a textbox (see Figure 3.8) one or more keywords that the Paragraph Retrieval will use to retrieve the most relevant paragraphs.

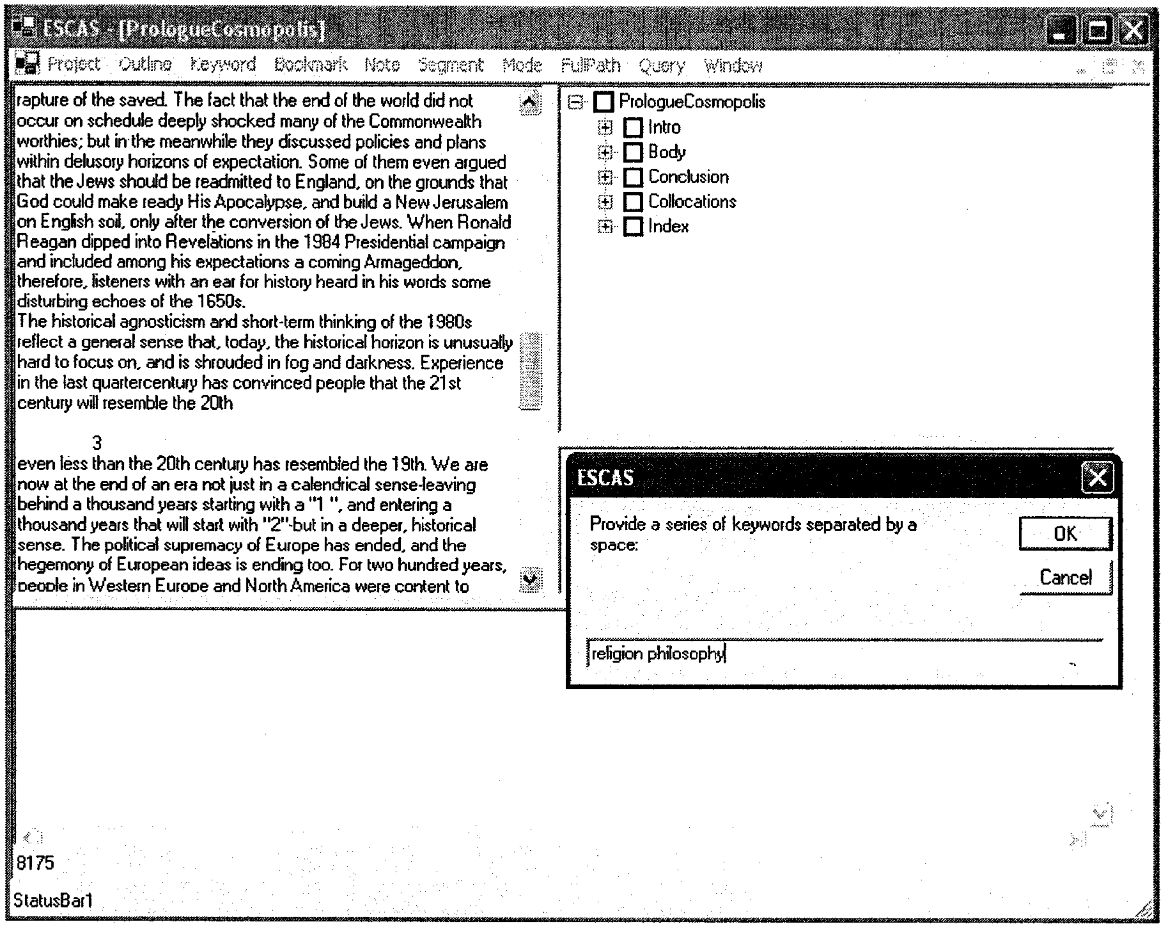

Figure 3.8 – Query box for Paragraph Retrieval

3.2.4. Query rewriting and reformulation

An important problem with statistical IR is that often very pertinent paragraphs containing terms semantically close to the query (such as synonyms) are not retrieved while noise is introduced in the retrieval set with paragraphs that are not pertinent. As an example, let's assume that a Philosophy term paper has to be written on "Potentiality and Actuality: Aristotle's reintroduction of unrealized possibilities and reaffirmation of becoming." A Query based on this subject would not include the word "Being" since that word is not among the words in the query. So the student has to somehow add this word to the query to improve the retrieval. Based on the new retrieval results, new words or phrases will suggest themselves and the user will start exploring the material with different queries. The benefit that can be derived is faster access to pertinent material within the text or exploring the material in unexpected but interesting directions.

3.2.5. WordNet

Query Reformulation can quickly turn short if the user does not have some help in getting more ideas on the subject. During the course of our research, the first way we found for the user to get help, short of using a thesaurus, was WordNet [Miller, 1995].

> *"WordNet® is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets."*[28]

The idea is to make available to the user semantically related words (synonyms, hypernyms, hyponyms, etc...) Their respective links allow the user to explore a certain domain and thus structure and enrich his Query Reformulation process. Figure 3.9 shows a small excerpt of the links available when a search is done on hyponyms (…"is a kind of") of the word 'Being' in WordNet using the Princeton Web online interface.

Sense 1
being, beingness, existence --
(the state or fact of existing; "a point of view gradually coming into being"; "laws in existence for centuries")

---

[28] http://wordnet.princeton.edu/

=> actuality -- (the state of actually existing objectively; "a hope that progressed from possibility to actuality")
=> reality, realness, realism -- (the state of being actual or real; "the reality of his situation slowly dawned on him")
=> fact --
(an event known to have happened or something known to have existed; "your fears have no basis in fact"; "how much of the story is fact and how much fiction is hard to tell")
=> reality --
(the state of the world as it really is rather than as you might want it to be; "businessmen have to face harsh realities")
=> historicalness -- (the state of having in fact existed in the past)
=> truth, the true, verity --
(conformity to reality or actuality; "they debated the truth of the proposition"; "the situation brought home to us the blunt truth of the military threat"; "he was famous for the truth of his portraits"; "he turned to religion in his search for eternal verities")
=> genuineness -- (the state of being genuine)
=> eternity, timelessness, timeless existence -- (a state of eternal existence believed in some religions to characterize the afterlife)
=> preexistence -- (existing in a former state or previous to something else)
=> coexistence -- (existing peacefully together)
=> subsistence -- (the state of existing in reality; having substance)
=> presence -- (the state of being present; current existence; "he tested for the presence of radon")
=> immanence, immanency, inherence -- (the state of being within or not going beyond a given domain)
=> ubiety -- (the state of existing and being localized in space)
=> ubiquity, ubiquitousness, omnipresence -- (the state of being everywhere at once (or seeming to be everywhere at once))
=> hereness -- (the state of being here in this place)
=> thereness -- (the state of being there--not here--in position)
=> thereness -- (real existence; "things are really there...capture the thereness of them"--Charles Hopkinson)

Figure 3.9 – WordNet example from the Princeton Web Interface

Figure 3.10 shows the corresponding implementation in ESCAS. The functionality provided in section 3.2.2.2 whereby the user can position himself on a collocation and include it into the current query with the pressing of a function key is also provided here. In the example below, the screen shown resulted from the user first submitting the word 'Being' and second, of the different senses provided, he selected the first which is "being, beingness, existence". Typically, he can now explore that sense. If the user wants to expand his query with that sense, he can bring the cursor to the top of the window and, with a function key, include the words "being, beingness, existence" in his query and resubmit it within his exploration process.

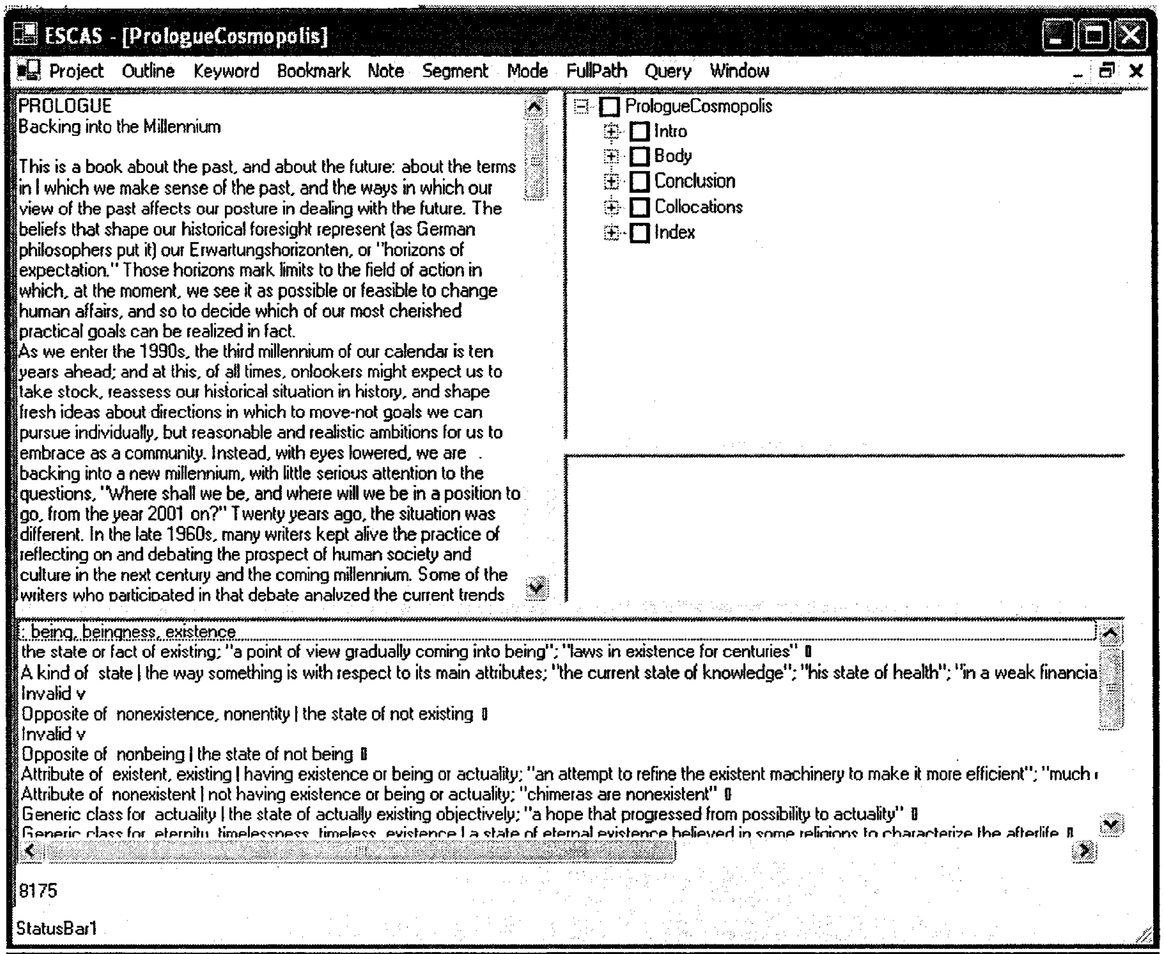

Figure 3.10 – Screenshot of the use of Wordnet

3.3. SCENARIO 2

It would be interesting to see how our paradigm would translate into a scenario different from scenario 1 proposed in Section 1.3

Julian opens the application. From the main screen he creates a new project called "The Crusades". A window is automatically created showing the root of a tree, the working outline. He immediately creates four keywords, the best he can do at this point, “Introduction”, “History”, “Analysis”, “Conclusion”.

He then opens the first of his four books, scanned into a text document. Two more windows are created; another tree containing the one-word and multiword indexes, the Reading Outline, and a text box with the actual text itself. He submits a query with the keywords ’Impact’, ‘Crusades’, ‘Middle’, ‘Ages’ to get the system to suggest some key passages. After reading the suggested paragraphs, he has a few doubts about his possible approach and decides to consult with his teacher.

Two days later, after his teacher has suggested some new keywords for his query, he decides to explore the other books with the paragraph retrieval function. In doing so, he realizes that the passages proposed from two of the books meet his immediate requirements such that he does not have to read them in detail. He will therefore concentrate on the other two books. He also has in mind, for these two books, some multi-word phrases; he plans to use the collocation facility to verify the context in which they are used.

He is now able to switch to the working outline and create two sub ‘User Text’ nodes to the Analysis node, ‘Religious causes’ and ‘Political causes’. He switches back to the top of the text box and starts reading. As he reads, like his sighted counterparts, he bookmarks, highlights, takes notes.

A bookmark is one keystroke away. Besides holding the position in the text, the bookmark is characterized by the first forty characters or so. This will give him a hint as to what the bookmark is all about when he reviews the list of bookmarks in a subsequent exercise.

To highlight, he creates a segment in the reading outline containing the text that would have been highlighted in a real book. The segment carries a pointer to its location in the text. So he can return to it at any time.

Some readers write notes on the very pages of books while others use yellow stickers. To do the same thing, he creates a note that also points to a place in the text. He can come back to that place to review the context in which he wrote his note.

While he is reading the book thoroughly, he will from time to time add additional keywords and sub keywords to the working outline, copy over some notes and segments. However, it's only after going through the four books that he will be in a position to review the three enriched reading outlines and populate what will be his final paper, the working outline. He can easily copy selected items from the reading outlines to the working outline. He eventually regroups, from the four books, all the references to a specific keyword in only one section.

While writing his paper he switches back and forth between the working outline and his paper. This allows him to easily work on the structure of the document, promoting sub sections into main sections, reordering or demoting sections; all this without having to perform extensive cut and paste operations. At the same time, he can have a view of what his paper would actually look like and do a cursive reading of it. (He will also need to read his document in an unfiltered mode, with the notes, highlighted text and bookmarks showing at the right places.) Once the content is complete, he generates a final Word document using an existing template if necessary. The table of contents is derived from the outline structure, the different levels of headings have the right fonts and the bibliographic entries for the six books are created in the Bibliography section.

	SCENARIO 1	SCENARIO 2
Feb 9th	Essay announced. Deadlines: Outline, March 15th Final, April 30th	Essay announced. Deadlines: Outline, March 15th Final, April 30th
Feb 28th	Starts reading	Starts reading
Mar 15th		Starts in depth study
Mar 21st	Starts in depth study	
Apr 5th		Starts outline creation
Apr 8th		Hands in outline
Apr 21st	Starts outline creation	
Apr 23rd		Teacher feedback obtained
Apr 30th		Additional books ready
May 5th	Hands in outline	
May 20th	Teacher feedback obtained	Hands in paper
May 27th	Additional books ready	
Jun 27th	Hands in paper	

Table 3 - Scenario comparison

Table 3 presents a timeline comparison between scenario 1 and 2. The purpose of this comparison is solely for illustrating the expected benefit of the system and is not part of an evaluation as such. The numbers for scenario 2 do not correspond to real experience.

They are judicious guesses given only for illustration purposes. Thus, discounting the time spent waiting for the books to be scanned and for the teacher's feedback, which would be the same in both scenarios, scenario 1 actually took 97 days to complete, while, after the intellectual exercise of transposition into scenario 2, the completion time would be 60 days, an improvement of 38 % with the paper handed in 37 days earlier.

Chapter 4

4. Technical Implementation of NLP features

In Chapter 3, we explained the functionality of ESCAS that we based on NLP techniques and more specifically, an information retrieval engine to retrieve the paragraphs that are the most pertinent to a user query. For this purpose, we navigate in WordNet and use its entries together with collocations to extend the query. To be able to properly put into context and evaluate this concept, we built a prototype. Its evaluation aspects are presented in the following chapter.

4.1. Document representation

In a computer, a text is represented as a string of characters. The first step in processing natural language is to break up this character string into units that will provide the basis for all our Natural Language Processing such as information retrieval and collocation finding. For our purpose, analysis at the word level is sufficient. Syntactic parsing is not required for implementing the NLP features we have chosen. As a result, every document that is opened in ESCAS needs to be tokenized into words. Several pre-processing issues have to be addressed before we can declare that our document representation is ready for our NLP solutions. The first is which words we keep for future processing. Grammatical words such as prepositions and determinants (for, to, the, etc.) do not make much difference for the meaning (semantics) in the text. These are also called stop words and are usually removed in tasks such as information retrieval. Secondly, we had to decide whether we needed to consider parts of speech tags because some solutions require that only nouns be considered while others require all the tokens. We opted for a full text solution because we considered that many parts of speech (e.g. verbs and adverbs) were indispensable for the type of text exploration a student would have to do with our software. Thirdly, we may want to consider two or more words as being one unit since they have the same root or stem. For example, 'transcendent' and 'transcending' can be assimilated to 'transcend'. Combining words having the same root into one stem is called 'Stemming'. The remaining words after removing the stop words are stemmed using the well-known and widely used Porter stemmer (Porter 1980). We decided to stem because the task of the user is to extract or summarize information around central themes. In that context, stemming seemed appropriate.

After tokenization, we create an inverted file (see Figure 4.1).

```
ideolog ideological 02 041 000600003600006648 190  034 000600003500006409 236
imagin imagination 01 025 000800004600008094 079
immin imminent 01 286 000400002200003732 314
import important 01 085 000100000800001616 242
improv improvement 02 115 000800004500007967 125  149 000600003600006648 190
```

Figure 4.1– Example of inverted file entries

For each stem (ex. ideolog), we keep a list of pointers to the paragraphs and sentences in which that stem occurs and to the position of the stem within that text. In the first entry in Figure 4.1, we can say that the stem ideolog has two occurrences in the text and the pointer values for each occurrence follow. We also keep one of the original words from which the stem was derived (ex. ideological) so that the user is presented with a word and not a stem whenever required by the user interface.

4.2. Indexing

Now that we have a list of words in the form of stems and their pointers in the text, it is straightforward to generate a text node for each index term and children nodes (as introduced in section 3.1.2 on outlines) to that text node with all the sentences where that index term occurs. The net effect is that we have created a user 'Index' where, instead of page numbers, the user can see the actual text excerpts which contains the entry through a set of pointers. By just pressing a key (Enter key in our case) on an occurrence, the user is taken to its exact location in the text. For the user, the index is handy because, in addition to giving access to the original text segment, the index can be used to scan through a list of sentences rather than to have to keep switching to the text from a list of page numbers. Section 4.4 on collocations will discuss the generation of multi-word terms.

4.3. Paragraph retrieval

The Information Retrieval technique we are introducing is to provide exploration facilities for a student when he is reading a document, by selectively presenting him with the paragraphs that are the most pertinent to the subject of his essay. The paragraphs in the text have been represented in the application by the inverted file. The application now needs a representation of the subject of the essay so as to be able to match it with the pertinent paragraphs. For this purpose, the user inputs a set of key words from the subject of the essay. This set will be called the "initial query".

In the vector model, every paragraph as well as the query is represented as a vector of weights with a vector dimension equal to the number of distinct index terms in all the paragraphs. The similarity between the query and a paragraph is then measured as the cosine of the angle between the two vectors. Based on such similarity, we can list the relevant paragraphs by order of relevance. The user can then consult the first "n" paragraphs according to her need.

An obvious candidate as a weight for a term in the vector is the raw frequency of each term in the paragraph with a weight of zero when the term is not found in the paragraph. For convenience, these frequencies are normalised with the maximum frequency in the paragraph. This metric is called the "term frequency" (tf) factor. There is another property of index terms that can be easily measured and is very useful. If a term appears in every single paragraph, it is not very useful for differentiating relevant paragraphs from irrelevant ones, while if it appears only in a few, such terms gain in importance. By taking the inverse of the frequency of the term in all the paragraphs, we have another useful metric. This metric is called the "inverse document frequency" (idf) factor.

As suggested by Salton [Baeza-Yates, Ribeiro-Neto, 1999], the retrieval of specific documents or paragraphs can be expressed as a clustering problem. The query is a requirement specification for retrieving a target set A from a collection C of documents. We must partition C into C1 and C2 such that C= C1 + C2 and C1 = A and C2 <> A. Intra-cluster similarity, or how well a term describes the contents of a paragraph, is measured by the frequency of each term within each paragraph, the term frequency (tf) factor. Inter-cluster dissimilarity, or 'how far a term is useful for distinguishing relevant paragraphs from non-relevant ones', is measured by the inverse of the frequency of that term in the system (all the paragraphs in the document), the inverse document frequency (idf) factor. Multiplying the two factors (tf*idf), we get an overall measure for the quality of set A.

In ESCAS, the number of query terms is expected to be relatively small since it is based on the subject of the thesis a user is writing. A reasonable number would be from 3 to 5 words. In the case of such small sizes for the query, Salton and Buckley [Baeza-Yates, Ribeiro-Neto, 1999] recommend that the query terms be weighted differently by using :

$$(0.5 + (0.5 * \text{normalized tf})) * \text{idf}$$

The tf*idf weighting scheme is standard in the field of IR; this is why we chose it to implement our paragraph retrieval module. It is also worth mentioning that IR is typically applied to entire documents while we are applying it to paragraphs.

4.4. Collocations

In section 3.2.2.2 we introduced the concept of collocations. Single words are often not enough to convey the full meaning that the user wishes to explore. Most of the time, the user is looking for material that is characterized by a combination of two or more words rather than just one. Collocations are of interest to us for two reasons. First, collocations are important for computational lexicography. Being a multi-word term combined together in a significant way, they are used to create a more useful index. Second, since the process of generating collocations uses a ranking scheme which integrates the importance of specific words within a document, this process produces a set of terms that characterize the documents. Several methods to identify collocations exist. In this section we will review the most important ones and justify the method we chose.

The simplest method is to count the number of times words occur together filtering those combinations that do not correspond to phrase patterns based on parts of speech (Justeson and Katz, 1995). However just the fact that words appear together with a high frequency does not mean that they are related. They could occur together by chance. Also, this method does not find combinations of words that are not adjacent. Since we are interested in more than fixed phrases, we had to consider other methods that would allow us to process windows of more than two words at a time and eliminate the chance factor.

Hypothesis testing is the classical way in statistics to determine whether an event is occurring by chance or not. It involves calculating the probability that words are associated by chance. The type of statistical test we choose will depend on the assumption we make whether the above probabilities are normally distributed or not. Church and Mercer (1993) have shown that the probability of word association is not normally distributed. Since the Pearson's chi-square test described in Manning (1999), contrary to the 't' test, does not assume a normal distribution, we chose the chi-square test.

Essentially, Pearson's chi-square test compares the observed frequency of pairs of words with the frequency expected if the two words were independent given by

$$X^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

where O_{ij} and E_{ij} are the observed and expected frequencies respectively in cell (i,j).

Below are examples of two bigrams, 'Actuality – Potency' and 'Object – Existent'.

The Chi-square for 'Actuality – Potency' bigram is 0.0649 calculated from the following word counts fom a text containing 527 words.

	'Actuality'	Other than 'Actuality'	Total
'Potency'	8	68	76
Other than 'Potency'	52	379	451
Total	60	467	527

Similarly the Chi-square for 'Object – Existent' bigram is 5.4768

	'Object'	Other than 'Object'	Total
'Existent'	3	40	43
Other than 'Existent'	8	476	484
Total	11	516	527

The critical value of a chi-square distribution at a probability level of $\alpha = 0.05$ is 3.841. The first bigram, with a chi-square lower than this threshold (3.841) is not retained for presentation to the user, while the second one is.

In this chapter we have described how the NLP features introduced in chapter 3 have been incorporated. The next chapter will now describe the evaluation of the system.

Chapter 5

5. Evaluation

The user community that we target with our application is very specific. One of the major difficulties in performing the evaluation of our system was to find blind students who are willing and have the time to learn a new computer based tool and subsequently evaluate the tool. Several steps were taken to find candidates, including contacting several Universities and CEGEPs and offering monetary rewards. It should be noted that blind students require more time to do their course work and have to catch up with the rest of their classmates. In this situation, it is hard for them to learn a new tool for evaluation purposes. As a result, the performance evaluation of our system, its learning and use by students will have to be integrated together over a period.

In cases where test resources are scarce, Nielsen [Nielsen, 1994] has successfully introduced an evaluation method that we adapt for this situation called 'Heuristic Evaluation'. One variation of this evaluation type is called 'expert evaluation with no heuristics'.

> *"A quick review by one expert (often without reference to specific heuristics) is usual before a user-based evaluation to identify potential problems."*
>
> http://www.usabilitynet.org/tools/expertheuristic.htm

Here we have one expert user using the system for a long time (over 2 years). This method is acceptable given the very specific scope of the thesis which is to "propose" a practical solution and prototype it. The disadvantage with this approach is that it can result in a single line of reasoning. To circumvent this disadvantage, we have invited another expert in the field to provide a "controlled" opinion. We will first describe the background of these evaluators and then discuss the characteristics of the application.

5.1. Evaluator 1

From the beginning of the project, a blind student named Didier was involved in the project. Didier is now a 2nd year Concordia University student enrolled in an undergraduate program in Philosophy and Religion. Since CEGEP, he had to write compositions for his course work. He was involved in the project since 2001, in incrementally determining the requirements of blind students, and has evaluated the prototypes at various stages of development.

5.1.1. Speed

From the beginning, our goal was to reduce the "time gap" between blind and regular students in writing compositions for a given course assignment. Our evaluator felt that after he started using this tool, it takes him about half the time to write an essay compared to the time he would take previously. The first effort at evaluating the application was to measure the overall efficiency of the system by comparing the time it takes a typical user to write a composition with and without the system. The difficulty here, is that any person (hopefully students, too) get better at a task each time they perform it. We cannot ask the same person to write a composition on the same topic twice. The second time will surely be faster and better. Also, as explained above, we do not have access to many student subjects. So, we settled for analysing the use of the system for real course work and compare it with the data we had previously compiled for another similar course work when the tool was not used. We must remember, however, that the two compositions are on different topics, of different lengths and graded by different professors.

Table 4 compares the two cases. To compensate for Didier's disability, in the last course in 2005, the professor gave him the topic in advance, while in the 2001 courses, the professor extended his deadline. Note that Didier is a full time student, but the activities below do not account for all his time.

	Regular Students		**Didier**			
	Due	**Days**	**Hand-in**	**Days**	**Diff**	**Ratio**
Without ESCAS						
2001	01-Mar 15-Apr	45	01-Mar 30-May	90	**45**	2.00
2001	09-Feb 30-Apr	81	09-Feb 27-Jun	138	**57**	1.70
With ESCAS						
2004	08-Sep 02-Dec	84	08-Sep 06-Dec	88	**4**	1.05
2004	08-Sep 13-Dec	95	08-Sep 19-Dec	101	**6**	1.06
With ESCAS						
2005	07-Feb 07-Mar	30	07-Feb 30-Mar	53	**23**	1.77
2005	16-Nov 09-Dec	23	02-Nov 12-Dec	40	**17**	1.74
2005	09-Nov 23-Nov	14	16-Nov 15-Dec	29	**15**	2.07

Table 4. Time comparison between regular students and evaluator

Since Didier had kept record of all his course work since 2001 together with the deadlines given for each course by the teacher, we were able to compare the time it took him to write compositions for his courses with the time regular students would take if they worked right up to the deadline. In 2001, Didier did not have the software, in 2002 and 2003, he used Version 1, and in 2004 and 2005, he used version 2 before the introduction of NLP techniques. Table 1 shows the data for years 2001, 2004 and 2005.

In 2001, it took Didier about twice as long to write a composition than regular students. If the professor gave 45 days to regular students, it would take him 90 days to achieve the same task. In 2004, the data seems to show a net reduction in time -he handed in his work about the same time as regular students. However, for 2005, it took him twice as long again. (The higher differences in 2005 compared to 2004, can be explained by more material having to be read and more courses taken simultaneously).

When looking at the difference between the time of a regular student and that of the evaluator, we see three clusters of data corresponding to each of the three years. ESCAS does show an improvement.

The differences seen in the case of year 2004 are much lower than those in year 2001, confirming our expectations. However, the differences seen in year 2005 are not progressively smaller as one would expect. In writing an essay the student has to perform two types of tasks in an interleaved fashion: one is to physically access the various resources and read them, and the second is to intellectually digest the materials read and create the report incrementally. The relative proportion of time spent in thee two tasks in solving a given assignment would vary. The ESCAS tool can assist in accessing the resources but not in the intellectual exercise. In the data shown in Table 4 the time spent in these two components are clubbed together. We describe this anomaly in looking at the ratio as follows: Suppose, for argument's sake, that a sighted user typically takes a constant 2 hours for physical access of the resources. The remaining time will be spent in the intellectual activity. This is shown in the two columns named 'Phys' and 'Intel' in Table 5.

	Regular Students			**Didier**			
	Days	**Phys**	**Intel**	**Days**	**Phys**	**Intel**	**Ratio**
Without ESCAS							
2001	45	2	43	90	47	43	**23,5**
2001	81	2	79	138	59	79	**29,5**
With ESCAS							
2004	84	2	82	88	6	82	**3**
2004	95	2	93	101	8	93	**4**
2005	30	2	28	53	25	28	**12,5**
2005	23	2	21	40	19	21	**9,5**
2005	14	2	12	29	17	12	**8,5**

Table 5. Time comparison of Table 4 revisited

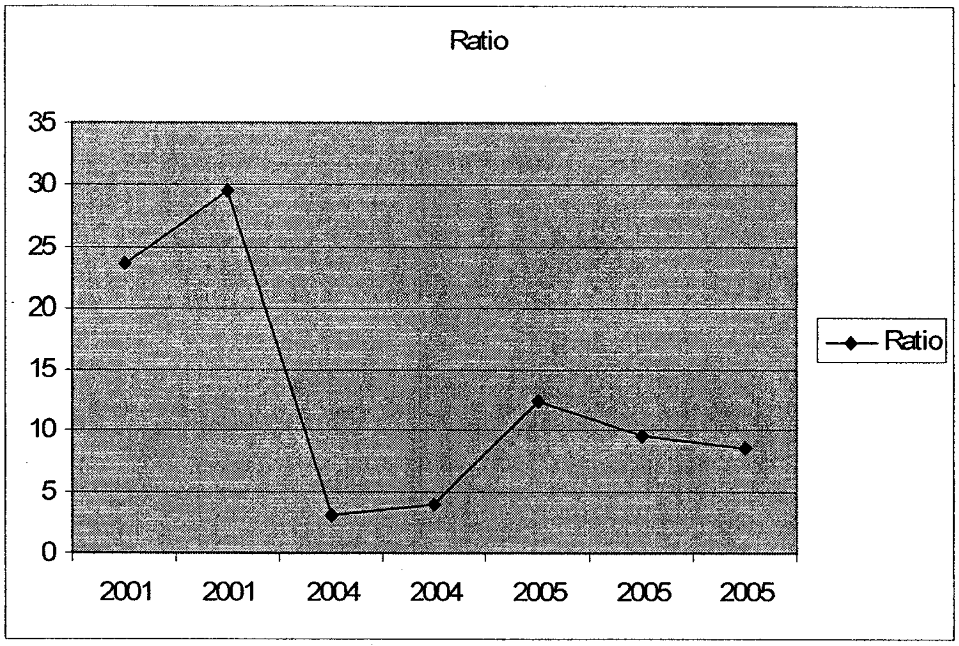

Figure 5.1– Physical access time of handicapped students expressed as a ratio of physical access time of regular students

If we assume the time spent in the intellectual activity to be the same for both the sighted and the blind, then the time spent in physical access by the blind varies. To obtain this in Table 5, we have subtracted from the total time reported by the evaluator, the amount equal to the intellectual time spent by the sighted. Recall the operations like browsing, skimming and glancing through are not easy for the blind and can result in much variations in time spent. The ratios shown in Table 5 are more reflective of the type of assistance that ESCAS can provide for the blind. These ratios are expressed as a graph in Figure 5.1 where the 2001 ratios are without ESCAS and the 2004 and 2005 ones, with ESCAS.

5.1.2. NLP tools

Paragraph Information Retrieval:

The experience of the evaluator with paragraph Information Retrieval was very positive. The paragraphs retrieved were really pertinent (precision) and helped the user zoom in on relevant material which he used to explore the text when the time came to write notes towards the creation of the paper. This exploration generated an appreciable savings in time. The algorithm chosen definitely serves its purpose.

Collocations:

The first attempt at creating collocations used a 10 word window and a threshold of 2. The process was extremely slow. A 3 word window and a threshold of 3.814 (as recommended by the text book) ran in reasonable time. The 20 entries with the highest chi square were retained for insertion in the outline. The cut off value of 20 was chosen arbitrarily.

From the word combinations produced, only some were pertinent. Moreover, the user found that bigrams (two words) are not enough as a hint when exploring the text, especially when compared with the full sentences that paragraph retrieval provides. Also, the user can be misled by "noisy word combinations". By the time the user knows enough of the material to filter the noisy bigrams, he is already at the composition stage and finds the multiword search and notes outline, as described above, much more effective.

5.1.3. Conclusion

Didier characterizes his experience as follows: *"If you know where you are going, ESCAS is a helpful and an indispensable tool. If you don't, it is like a dead weight to carry. More concretely, you must first have a good idea of your topic and your material before creating an outline and fleshing it out into a paper."* In the beginning he used to retrieve many documents and paragraphs, then spend a lot of time reviewing it all. He was falling into the familiar trap of beginning students who would highlight the whole book.

Today, Didier claims that it takes him about half the time to write a composition when using the system compared to the time he would take previously. He can skim literature more quickly and can concentrate on more creative thinking than he could without the system. All in all, Didier has been using the system every semester since we offered it to him and he considers it a necessary tool for writing. He contends that from paper to paper his methodology changes. But this would not be possible without ESCAS. He says he progresses by exploring what he can do with ESCAS. Just like anybody would progress while using software (Excel for example) and change the way he does things. There is a synergy between the way he uses it (the methodology) and his skill with the software (his actual use of it).

5.2. Evaluator 2

The second evaluation was performed by Dr. Leo Bissonnette who is a visually handicapped person. Leo has a long experience with disabled students as the coordinator of Concordia University's Office for Students with Disabilities. His 15 years teaching experience in Sociology covers 200 and 300 level courses. His combined expertise makes him a candidate of choice for performing a "controlled" evaluation of ESCAS from both a **theoretical** and **practical** point of view. He has had the chance to use ESCAS on two projects. We subsequently interviewed him. The following paragraphs will summarize his feedback. The text of the interview is reproduced in full in Appendix I. His feedback made it possible for us to validate both the relevance of the problem described in this thesis as well as its solution.

5.2.1. Relevance of the problem addressed

In Section 1.2.2, we identified four major tasks into which we could subdivide the term paper writing: Research, Analysis, Outlining and Composition. We used the same categories to structure Leo's evaluation.

5.2.1.1. Research

In this first stage of the process, a student gets acquainted as quickly as possible with all the material at his disposal through **reading** and **skimming** for two reasons. First to have a basic idea of what he will talk about, and second, to be able to choose from this material what he will have to study in detail in the next task.

Reading

Leo constantly mentions three groups of special needs students, the print impaired, the mobility impaired and the learning disabled. All these groups have said that they try to recall materials from memory. For example, students with learning disabilities have often told Leo that they rely for the most part on lecture notes. They try to avoid reading and stay away from courses requiring heavy reading. Those with mobility impairments have under utilized taped/recorded texts in the past and have struggled with reading. This strategy of shying away from reading and relying on memory, as acknowledged by the students themselves, falls short of what is required in university and graduate courses. As a result graduation rates are relatively low.

Skimming

According to Leo, skimming for content is difficult. One reason for this is that finding headings is a key to skimming and traditional reading methods have made such skimming impossible. Leo gave two cases where skimming is very important. A student should ideally prepare for a new topic and an upcoming lecture by quickly reviewing the night before. The same could also be said for a final review before an exam. However, skimming just 'doesn't happen'.

Leo also makes reference to those students who do their own scanning of documents with Optical Character Recognition (OCR) software either to have the document read out to them or saved as a text file for eventual reading. Those students still need to find ways to quickly work through text for major ideas.

5.2.1.2. Analysis

The analysis stage requires that students study in detail the material they have selected in the previous stage. This stage should result in a set of notes and excerpts to be used as input into the next stage. To characterize the requirement at this stage, Leo has emphatically introduced the concept of 'Active Reading' as a key concept for disabled students. We will try to describe this concept using various definitions from academia.

> *"Effective readers are ACTIVE readers. As they read they:*
>
> - *know what they are looking for and how to find it*
> - *relate new knowledge to old knowledge*
> - *make patterns and connections*
> - *ask questions about the text.*
>
> *Thus effective reading involves many of the features of critical reflection."*

http://www.une.edu.au/tlc/alo/active4.htm
University of New England (Australia, 2006)

> "'Active Reading' *is reading that involves the active use of the cognitive system. Can be contrasted to reading that simply involves mouthing words without active comprehension of what the words mean."*

http://www.cognitive-aptitude-assessment-software.com/Glossary/ActiveReading.html

> *"It is very important to be an ACTIVE reader as this will help you retain information in a text and help you make the right kind of notes – it is essentially reading for a purpose rather than just browsing."*
>
> University of Southampton, 2003

> ***"Strategy 3: Active Reading***
> *When you are reading a document in detail, it often helps if you highlight, underline and annotate it as you go on. This emphasizes information in your mind, and helps you to review important points later.*
>
> *Doing this also helps to keep your mind focused on the material and stops it wandering."*
>
> http://www.mindtools.com/rdstratg.html

Leo is quite clear on that subject. *"For most students the basic notion of actively summarizing and taking notes while reading just doesn't happen at all... The visually impaired have historically used tapes and very few of them have read actively...Students with disabilities need to be pushed to do more* (Active Reading)*."*

5.2.1.3. Outlining

An additional challenge besides actively writing notes is keeping track of them. According to Leo, keeping track of summaries and notes and making linkages between sources is a big concern for students, especially in the context of comprehensive exams preparation. They often have to reread specific material to develop links between ideas and to note levels of meaning and connections. Besides, students are often asked by professors to demonstrate their ability to synthesize materials from diverse sources. This is especially true in the social sciences and history.

Leo has found that in the above activity *"many students will fail to work efficiently. In fact they will give up."* His own experience as a blind doctoral student was that "*keeping track of where I had my notes for easy retrieval was quite a task. I found myself using folders on a hard drive and the find feature. But at times even this proved very time consuming.*"

5.2.2. Evaluation of the solution

The following is a compilation of Leo's comments about his use of ESCAS:

5.2.2.1. Positive comments

1. The software described in this project gets us past the barrier that many students who are print impaired meet, that of being underread or passive readers or not having a strategy for reading economically and efficiently. *"Skimming for common themes and ideas is greatly improved. One then quickly gets down to exploring in greater detail subthemes and ideas. Writing then becomes a more creative process as outlines are used to push the reading/writing process along in a dialectical fashion"*.
2. The logic is sound and as the user works with the application, the approach becomes clearer and more intuitive. So from a user perspective, the design is solid!
3. The ease of moving items makes the program especially attractive. It certainly promotes active reading/thinking.

4. The software has real benefits for those with other disabilities--especially learning disabilities.
5. The concept of 'Working Outline' is truly the benefit here. Getting people to work through a process and having a document is the payoff in terms of time and efficiency--especially in the final writing and editing stage.
6. The software enables users to see writing as a creative process as opposed to being an unwelcome task.
7. The proposed software is a real solution for keeping track of notes and their easy retrieval.
8. The software works nicely with JAWS in the HTML version of the documents for finding such things as headings. This feature promotes collaborative work with others which is becoming very common both in academic and work situations.

5.2.2.2. Shortcomings and suggestions

1. The software has problems working with JAWS, the screen reader which is a popular tool used by the visually handicapped. The voice stopped several times and the user lost his place in the text.
2. The software does not differentiate between novice, intermediate and advanced users.
3. Exercises in training tutorials of different levels should be built in order to get people to use the software.
4. The following functions would need to be explained and promoted as learning strategies for both reading and note taking and are more appropriate for advanced

users. "*Now we have the possibility of showing students how to read differently in order to do different things.*"

   a. The user generates a segment to mimic highlighted text to which he can return to explore its context if required,
   b. The user can write notes tied to a specific position in the text. Pressing Enter on a text node will return the user to that position. This allows the mimicking of yellow stickers or handwritten notes in a book.
   c. Paragraph retrieval. The learning disabled population would certainly have more difficulties using paragraph retrieval.
   d. The use of WordNet. Likewise, the learning disabled population would have more difficulties using WordNet.

5. The inclusion of automatic recognition of headings and sub-headings is strongly suggested.
6. Leo mentioned the possibility of evaluating the Cornell method [Santrock, Halonen, 2005] for taking notes within the context of this application. This method divides the note paper into three sections: notes in the right column, key words and questions in the left column, and a summary at the bottom of the page.

5.2.3. Speed

In simulating two small projects Leo estimated that he saved 40% of his time by using the application. His collection of key points was significantly reduced and he was able to finish both projects with a second draft.

5.3. Conclusion

Leo's conclusion is that ESCAS is a useful and usable application that definitely meets the objective of time reduction for the print impaired when writing compositions. It could apply to other impairments such as Dyslexia and Learning Disabilities. He was very enthusiastic about it and would recommend it to his peers at various local colleges and Learning Centres (Dawson, Vanier, Marianopolis and Cégep du Vieux-Montréal are examples) in the spirit of "*pushing the idea that reading and writing are active processes and encouraging learners who have not historically had this opportunity*".

Chapter 6

6. Conclusion and Future Work

The prototype system that we built can be seen as an information probing and gathering environment that offers features based on composition tools to manipulate objects such as outlines, bookmarks, stick notes and based on NLP techniques. Based on the evaluation, we can conclude that, for at least two evaluators, we have succeeded in making a change and one of them has adopted this tool for all his work and considers the tool indispensable. However, this experience has opened us several important opportunities for future work that need to be explored before we can say that we are truly successful.

6.1. Further evaluation

Our current evaluation suffers from several problems. The number of evaluators and the depth of evaluation require attention. Future work should definitely include a more extensive evaluation of the system with several blind and non-blind students from a post-secondary school and a comparison of their use of the system. However, how truly formal can evaluation be? We simply cannot give the same composition twice to the same subject, and expect to measure only the utility of the system. We need to evaluate qualitatively the usability of the system and devise a set of experiments with a qualitative metric to show scientifically where and how ESCAS is useful. For example, we could capture automatically the use of the system through the keystrokes of the user and log the functions used and the frequency of that use to identify the usefulness of the different features.

6.2. Natural Language Processing enhancements

When experimenting with collocations, we had to use bigrams in a 3 word window for the system to run in a reasonable time. Also, the threshold for the chi-square test was 3.814. The fourth parameter in our experimentation was the use of only two expert evaluators who did not find collocations useful for exploring text. There is definitely room for further research here by gradually increasing the different variables and involving more than two evaluators. For example, using 4-grams in a five word window may yield more meaningful phrases of 4 words instead of two with a higher relevancy.

We also propose to evaluate the different API's for Automatic Summarization in greater depth to see if they can be easily integrated into our solution. In doing so, we would make available to our user richer and more complete text analysis and Document Understanding facilities. With reference to section 2.1, our application could have the following characteristics:

- It would process single text documents from books or articles on the Web.
- We would restrict ourselves to a specific domain such as History, Religion, Psychology or Philosophy.

- The output would be of the informative and indicative type, centered on specific user interests.
- Several compression rates would be available as user options.
- The user would have the choice between text-driven functionality to establish the relevance of documents and query-driven functionality for more precise information probing.
- Extract based software would be chosen.

6.3. Adapting to the DAISY standard

The DAISY standard used to encode text and audio in the same document was introduced in Chapter 2 and we refer the reader to this chapter for more details. More and more textbooks will be produced using this standard. How does this multimodality affect our paradigm? How can we reproduce the same highlighting facilities on audio as well as on text? How do we exploit and integrate voice control in that context? Another feature would be to make text residing in digital textbooks available through ESCAS.

6.4. WordNet, Ontologies and Intensional Navigation

In section 3.2.4 on Query rewriting we suggested to use Wordnet as a source of information to be used for Query reformulation. A domain specific ontology could also be useful to the user as an additional source of information for the same purpose. In fact, several ontologies with possibly different levels of expressiveness could provide a rich environment. Ontologies are a powerful way to represent the knowledge of a domain. Navigation in those ontologies will require 'intensional navigation'. Intensional navigation is a term used mainly in the context of navigation in large information spaces, and more specifically in hypermedia [Franconi, 2000]. The study of navigational aids will become more and more relevant as those spaces become larger and access to information in large hypertexts and the growing 'semantic web' will require facilitation.

6.5. Final word

At the outset of this project, we realized that we were harnessing ourselves to a difficult but important task; that of making a difference in the study life of blind CEGEP and University students by assisting them in writing compositions. As we progressed, we found we were uncovering a new but definite need in this community of people, especially when comparing what had been done so far. To fill this need, our line of investigation was to use composition tools and Natural Language Processing techniques. To be able to make these techniques available effectively to the user, we proposed a new paradigm within which we tried to build a solution and provide a non-trivial and solid base for future developments.

We have eventually realized that our effort joins that of the mainstream. Our problem is that of finding the right information within an intimidating mass of data. For the public at large, tools like Google help them to zoom in on the right article or document. For our category of users, we can easily transpose into finding the right sentence, the right key phrase within an equally intimidating amount of data, intimidating because the access to this data is limited by our 40 character or 7 word window, comparable to studying only with a cell phone for a regular student. This opens up further questions and opportunities. It is in fact only beginning.

We hope that our contribution will be useful and that it will help visually handicapped students to better exploit their potential.

BIBLIOGRAPHY

[Baeza-Yates, Ribeiro-Neto, 1999] Ricardo Baeza-Yates and Berthier Ribeiro-Neto.Modern Information Retrieval. ACM Press and Addison Wesley, 1999

[Cowan 2001] Cowan, N. (2001) The Magical Number 4 in Short-term Memory: A Reconsideration of Mental Storage Capacity. *Behavioral and Brain Sciences* 24 (1).

[Edmundson, 1969] H.P. Edmundson. New methods in automatic extracting. *Journal of the Association for Computing Machinery* 16(2): 264-285. Reprinted in Advances in Automatic Text Summarization, I. Mani and M.T. Maybury (eds.), 21-42. Cambridge, Massachusetts: The MIT Press, 1999.

[Franconi, 2000] Enrico Franconi. *Knowledge Representation meets Digital Libraries.* European Research Consortium for Informatics and Mathematics (ERCIM) Workshop Proceedings of the First DELOS Network of Excellence Workshop on "Information Seeking, Searching and Querying in Digital Libraries" Zurich, Switzerland, December, 11-12, 2000

[Lester, 1996] James D. Lester. *Writing Research Papers: A Complete Guide.* New York, USA: Harper Collins College Publishers, 1996.

[Mani, 2001] Inderjeet Mani. *Automatic Summarization.* Philadelphia, USA: John Benjamins BV, 2001.

[Manning & Schütze, 1999] Chris Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, MA..

[Marcu, 2000] Daniel Marcu. *The theory and practice of discourse parsing and summarization.* Cambridge, Massachusetts: The MIT Press, 2000.

[Miller, 1995] G. Miller. *WordNet: a Lexical Database for English.* Communications of the ACM, 38(1), 39-41.

[Nielsen, 1994] Nielsen, J.. Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*. John Wiley & Sons, New York, NY, 1994.

[Saggion, Lapalme, 2002] Horacio Saggion, Guy Lapalme. Generating Indicative-Informative Summaries with SumUM. *Computational Linguistics*. Volume 8, Number 4, December 2002

[Santrock, Halonen, 2005] John W. Santrock, Jane S. Halonen. *Your Guide to Collge Success: Strategies for Achieving Your Goals, Concise Edition*. Thompson Wadsworth, Belmont, California, 2005.

[Shea, Whitla, 2001] Victor Shea and William Whitla. *Foundations : CriticalThinking, Reading, and Writing*. Toronto, Canada: Prentice Hall, 2001.

[Sparck Jones & Willett, 1997] Karen Sparck Jones and Peter Willett. *Readings in Information Retrieval*, San Francisco, California, Morgan Kaufmann, 1997.

[Sparck-Jones, 1999] Karen Sparck-Jones. Introduction to Text Summarization. In Inderjeet Mani and Mark Maybury edts., *Advances in Automatic Summarization*, pages 1-12, The MIT Press, 1999.

[Turney, 1999] P. Turney, *Learning to Extract Keyphrases from Text*, ERB-1057, NRC-41622, Institute for Information Technology, National Research Council Canada, February 17, 1999.

[Willis, 1977] Hulon Willis. *Writing Term Papers : The Research Paper & The Critical Paper*. New York : Harcourt Brace Jovanovich Inc., 1977.

APPENDIX – An interview with Dr. Bissonnette

A. Based on your experience, how far are the following problems true and relevant:

1. The amount of material to be read is often a challenge for a sighted student, let alone for handicapped students.

This is particularly true for undergraduate students in the faculties of Arts and Science--especially courses designed to be heavy reading courses. Adding on the level of graduate studies, also adds significant reading loads/requirements. For students preparing for comprehensive exams this is a big concern--that is, keeping track of summaries and notes; making linkages between sources.

Many students who are print impaired due to a visual impairment are underread. They are in fact passive readers; not active readers. They do not in fact have the strategy for reading economically and efficiently. The software described in this project gets us past this barrier.

2. Students with visual handicaps, dyslexia and learning disabilities have reading difficulties that go from very mild to very severe due to lack of reading and summarization skills as well as physical constraints. They tend to read less than other students.

This is true. The visually impaired have historically used tapes and very few of them have read actively--that is summarizing materials and synthesizing materials from various sources.

Students with learning disabilities have often told us that they rely for the most part on lecture notes. They try to avoid reading and stay away from courses requiring heavy reading.

Those with mobility impairments have under utilized taped/recorded texts in the past and have struggled with reading.

All the above groups have said that they try to recall materials from memory, but acknowledge that this strategy falls short of what is required in university and graduate courses. As a result graduation rates are relatively low.

3. Seeing everything through a 40 character window for Braille users, or a window of less than 10 spoken words for speech users creates a major challenge, that of not having any overview at any level.

Braille displays of any size are interesting devices because on one hand they provide more of an equivalent for the eye/brain connection for the sighted reader by giving the blind user a hand/brain connection. In other words, when the reader has his hand on a word it is always present and the cognitive processing is different than that when speech is the only medium. When words are spoken to the reader it is a transitory process. If the reader needs to hear the words again keystrokes are required to refresh the speech of the specific words.

Braille displays have as a disadvantage the need for keystrokes to advance the display. Newer displays do now have automatic reading features, but the user must spend time experimenting with reading speed rates.

4. Skimming is impossible for a blind person. What appears to be substantial for a regular student, in terms of the amount of material to be read, becomes massive for a visually handicapped student.

Skimming for content is difficult. Finding headings is key to such a process and traditional reading methods have made such skimming impossible. The new generation of digital readers offer some hope here--assuming that the text is marked for such things as headings and subheadings.

Even those who do their own scanning still need to find ways to quickly work through text for major ideas.

Ideally a student should prepare for a new topic and an upcoming lecture by quickly reviewing the night before the key points in a chapter to be read. The same could also be said for a final review before an exam. Skimming doesn't happen and any tool that will promote the taking of notes and the generation of headings, key phrases will greatly increase this possibility.

5. Reading material several times is often required where reading only once would be sufficient for a sighted reader.

I would suggest that this is true. However, I would add that now we have the possibility of showing students how to read differently in order to do different things. For example, after reading several similar articles on a topic it is often helpful to reread a specific article to develop links between ideas and to note levels of meaning and connections. Again, we get back to active reading. Reading at times needs to be iterative and students with disabilities need to be pushed to do more of this. Often students are asked by professors to demonstrate their ability to synthesize materials from diverse sources. This is especially true in the social sciences and history.

6. Note taking is an art in itself. The classic index card method facilitates sorting of notes but is tedious and hardly used by sighted students. Taking notes can present additional challenges to students with certain handicaps.

This is true and I would suggest that for most students the basic notion of actively summarizing and taking notes while reading just doesn't happen at all. Based upon personal experience in recent years as a doctoral student I found it liberating to take notes. However, keeping track of where I had them for easy retrieval was quite a task. Jacques proposed software is a real solution here!

7. Putting order and structure in the notes is a tedious process. There is no way to structure them *easily. Sometimes several physical documents have to be manipulated. If material has to be reviewed again and again based on a reference made in the notes, a book or document may have to be practically "read" several times.*

The retrieval and linkage between common ideas and books containing the information is where many students will fail to work efficiently. In fact they will give up.

Again, based upon my own recent experience, I found myself using folders on a hard drive and the find feature. But at times even this proved very time consuming.

B. How far does each of the following elements of our proposed solution help towards solving the above problems?

1. Since the purpose of the exercise is to produce a structured document, the basic architecture is centered on outlines grouped into one project.

The logic is sound and I found that as I worked with it the approach became clearer and more intuitive. So from a user perspective, the design is solid!

2. An outline consists of items and sub items, also called nodes, organized into the form of a tree data structure. The nodes can be bookmarks, segments, notes and user text. They can all be copied, promoted to a higher level, demoted to be the sub item of another item or deleted.

Again, what I said above holds true. The ease of moving items makes the program especially attractive. It certainly promotes active reading/thinking. We'd need to promote the program in conjunction with learning strategies for both reading and notetaking. The software has real benefits for those with other disabilities--especially learning disabilities.

3. The 'Working Outline' corresponds to the final document being created.

This is truly the benefit here. Getting people to work through a process and having a document is the payoff in terms of time and efficiency--especially in the final writing and editing stage. One of the benefits of word processing is that over time the cognitive approach to how work gets done is key. I am convinced that the software here is a liberating one because it will enable users to see writing as a creative process as opposed to being an unwelcomed task.

4. The 'Reading Outlines' correspond to the input documents and serve to capture their structure through bookmarking, taking notes and identifying parts of the text that could be used for quotes in the final document.

Again, many of my comments made above, would also apply here. The benefit is that users are enabled to read actively using easy-to-use tools.

5. When a search is required in either the document area or an outline, the user can bring up a textbox to enter the search string. The search can be easily repeated from the last occurrence to the end of the document.

I had a few problems here working with JAWS, my screen reader. The voice stopped several times and I lost my place. This may be a problem more with JAWS and the script file supporting it. The concept of the search and its functionality give the software power, making finding easier and pleasant.

6. An outline can be exported to a text file that can be used as a starting point for a final document. The outline can also be exported into an HTML file with links corresponding to the main and sub headings. This is particularly useful for collaborative work with other students. Preliminary notes as well as advanced work can be shared.

Yes! It also works nicely with JAWS in the HTML versions for finding such things as headings.

Collaborative work with others is becoming very common both in academic and work situations. This feature will make it an attractive software package with potential commercial benefits.

7. Students with no visual handicap use several artefacts to generate bookmarks like 'marking the book' with a highlighter. When they need to go back to review these places, they can skim through the book. In our solution, when the user wants to ensure he can subsequently return to where he is in the text, he can generate a bookmark. A segment is a fragment of text from a document that mimics highlighted text. A segment allows the user to return to the position where the segment was extracted and explore once again the context of the segment if required.

This is an important function. Here we would need to explain as a strategy why we would do this. Once done, I am convinced that this feature would be used a great deal by advanced users.

8. The user can write notes tied to a specific position in the text. Pressing Enter on a text node will return the user to that position. This allows the mimicking of yellow stickers or handwritten notes in a book.

Again, we get back to active writing and notetaking. This is a tool we'd need to promote and explain as a strategy; then build in exercises in a training tutorial in order to get people using it.

9. Paragraph retrieval suggests to the user the places where he would start reading that would be the most relevant to the subject being treated. The paragraphs retrieved are pertinent and help the user zoom in on relevant material which he can use to explore the text when the time comes to write notes towards the creation of the paper.

Again, promoting active reading/writing/retrieval of such points needs to be taught to users. I am convinced that a tutorial would further enhance the strategy side of this software.

10. Collocations provide help to the user when the latter wants to get acquainted as quickly as possible with a document.

Such an "overviewing" is key and will further promote active work in a project by the user.

11. The user can add additional words to his query to improve the retrieval. Based on the new retrieval results, the user can start exploring the material with different queries in unexpected but interesting directions. The user can position himself on a collocation and include it into his query.

This is key to promoting reading as an iterative activity. Very essential at project levels.

12. WordNet makes available to the user semantically related words (synonyms, hypernyms, hyponyms, etc...) Their respective links allow the user to explore a certain domain and thus enrich his Query Reformulation process. The user can expand his query from where the cursor is so he can resubmit that query within his exploration process.

I see the above as something for more advanced users. Certainly the learning disabled population would have more difficulties here. Again, a tutorial with exercises would help here.

13. This prototype generates an appreciable savings in time when writing a paper.

In simulating two small projects I estimate that I saved 40%. My collecting of key points was significantly reduced. I was able to finish both projects with a second draft.

14. The user can skim literature more quickly and can concentrate on more creative thinking than *he would without the system.*

Skimming for common themes and ideas is greatly improved. One then quickly gets down to exploring in greater detail subthemes and ideas. Writing then becomes a more creative process as outlines are used to push the reading/writing process along in a dialectical fashion.

C. Would you recommend this software to peers at various local colleges (Dawson, Vanier, etc.)?

I would. I'd ideally like to see people in the Learning Centres shown the software. Again, pushing the idea that reading and writing are active processes and we need to encourage learners who have not historically had this opportunity, I would suggest that these institutions would be interested in the software.