Winsight: Towards Completely Automatic Backtranslation of Nemeth Code

D. Gopal[1], Q. Wang[1,2], G. Gupta[2], S. Chitnis[1], H. Guo[3], and A. Karshmer[4]

[1] Logical Software Solutions, Plano, TX
[2] University of Texas at Dallas, Richardson, TX
[3] University of Nebraska at Omaha, Omaha, NE
[4] University of San Francisco, San Francisco, CA

Abstract. We present the Winsight system, a Windows-based software system for completely automatic translation of Nemeth Braille code to LaTeX. The Winsight system takes hard copy Braille input containing Mathematics (written in Nemeth Braille code) and text (written in contracted Braille) via a scanner, performs image recognition and analysis of the scanned file to generate the ASCII Braille file, automatically separates Nemeth Braille coded expressions and contracted Braille text, backtranslates them to LaTeX math expressions and LaTeX text respectively, and produces a print output file in pdf format containing the result of backtranslation. The Winsight system comes with tools that allow users to manually intervene during each step, if they desire, to fix any errors reported by the system or seen by the user. In this paper we give an overview of the Winsight system.

1 Introduction

Visually Impaired individuals "read" and write using Braille. Various encodings have been used to make Braille writings more succinct and efficient to read (for example, contracted Braille codes). Braille-based notations have also been developed to encode Mathematical expressions, for example, the Nemeth Braille code for mathematics [4]. Visually impaired students in the United States (as well as in other countries such as New Zealand, Canada, Greece, India, etc.) typically learn the Nemeth Math notation to "read" and write Mathematics. However, when they take a Mathematics course, the following problem arises. The instructor (who is typically sighted) prepares all the course material (such as home assignments, exams) using LaTeX, a popular word processing package used for typesetting Mathematics [7]. The visually impaired student cannot access this homework; the homework has to be read out to him by someone. Next, the visually impaired student might write the answers to problems in the homework using Nemeth Math notation. The solution is unreadable by the instructor since he/she neither knows the Nemeth notation nor Braille. This communication gap between visually impaired students and their sighted instructors is a major problem that hinders Mathematics and Science education for visually impaired

C. Stephanidis (Ed.): Universal Access in HCI, Part III, HCII 2007, LNCS 4556, pp. 309–318, 2007.

individuals. In this paper we describe our automatic solution for bridging this communication divide.

To illustrate the problem, consider how visually impaired students communicate with their sighted instructors. For example, in response to a question, a visually impaired student may write the following proof using contracted Braille and Nemeth Math Braille code (we use the ASCII equivalent of the Braille code called ASCII Braille; ASCII Braille code is normally used to represent Braille on a computer):

```
,pro( ,let ;a & ;b be two po9ts 9 ! 9t]val )
a "k b !n "! is "s x 9 (a, b) )
    f'(x) .k ?f(b)-f(a)/b-a#,
b f'(x) .1 0 = all x 9 (a, b) ,so
     ?f(b)-f(a)/b-a# .1 0,
s9ce b-a .1 0 x foll[s t f(b) .1 f(a)
```

As the reader can notice, this is hard to read without knowledge of the two Braille codes (contracted Braille and Nemeth code) involved. The above proof stands for:

Proof Let a and b be two points in the interval with $a < b$ then there is some x in (a, b) with

$$f'(x) = \frac{f(b) - f(a)}{b - a}$$

But $f'(x) > 0$ for all x in (a, b) So

$$\frac{f(b) - f(a)}{b - a} > 0$$

Since $b - a > 0$ it follows that $f(b) > f(a)$

One way to solve this communication problem is to build automatic translators between LaTeX and Nemeth Math notation so that the homework assignment can be automatically translated from LaTeX to Nemeth code for the student to "read," while the homework answers prepared by the student in Nemeth code can be automatically translated from Nemeth Math to LaTeX for the sighted instructor to read (the process of translating Nemeth code to LaTeX is called *backtranslation*). Building the translator and backtranslator, however, is a difficult and time-consuming task.

Building a translator that automatically translates LaTeX to Nemeth Math is relatively easy, as LaTeX is a recently designed notation, and has a *context free grammar* (which means that LaTeX documents can be easily parsed automatically by a computer). Indeed, a translator from LaTeX to Nemeth Math is commercially available from MacKichan Software as part of their Scientific Notebook System [8] (currently marketed by Duxbury Systems). In contrast, the Nemeth Braille Math notation is quite complex, since it is designed to be human readable. As a result, the backtranslation problem is considerably more difficult and in the past has been considered as unsolvable [6]. The problem of backtranslation becomes harder if Nemeth code is embedded within contracted Braille text. In fact, most mathematical documents will contain both Math expressions and English text.

Assuming that a backtranslator system is available, an additional problem arises. The backtranslator will take the Math document written by blind individual as input, which has to be provided in an electronic form, such as ASCII Braille. Typically most blind students write mathematical documents using a mechanical Brailler (e.g., Perkins Brailler). Electronic Braille devices (such as *Notetakers*, with refreshable Braille display) which can produce an electronic Braille document are not used even if a blind person has access to them. This is because such Braille displays can only display one line at a time, and while solving mathematical problems, frequently the writer needs to refer back to several of the previous lines, especially, if spatial structures (e.g., matrices, determinants) are used. The mechanical Braillers only produce a hard-copy document, there is no electronic file produced. The problem then one faces is the conversion of hard-copy Braille documents into an electronic format that can be fed to the backtranslator.

In this paper we report on our work on the Winsight system, a system that *completely automates* the backtranslation process. The system consists of 3 modules: the OBR module, the *separation module* and the *backtranslation module*, all developed by us. Hard copy Braille documents produced by mechanical Braillers and containing both Math (written in Nemeth Braille Code) and English text (written in level 2 Braille) are scanned on a standard scanner, and the image file obtained automatically processed by the Winsight OBR module to recognize the Braille characters. The Braille characters recognized are stored in a file in the ASCII Braille format. This file is further processed by the Winsight separation module to automatically detect parts that are written in Nemeth code and those that are written in contracted Braille. This prior separation of Nemeth and contracted Braille components is important, because, in general, it is difficult to automatically distinguish between the two codes (for example, the symbol '+' means different things in the two codes). The separation module will insert the string `;;` at the beginning and the string `;'` at the end of each Nemeth math expression it recognizes. Text outside of these begin-end symbols is then assumed to be in contracted Braille. This annotated file is then fed to Winsight's backtranslation module, which produces a LaTeX file that can then be rendered in the pdf format to be read by a sighted person (e.g., a course instructor).

An additional feature of the Winsight system is that the user can intervene after any of the intermediate steps to fix problems. Thus, the user can edit the image file after it has been scanned. He/she can edit the *blob* file obtained after image processing, as well as the ASCII Braille file obtained after the Braille character inferencing has been done. Finally, the user can also edit the result of automatic separation of Math and contracted Braille text. This user intervention is needed because typically hand typed Braille documents have many errors of spacing (e.g., paper may have moved during typing), the Braille character may have been intentionally erased by the user either by backspacing or by typing the '=' Braille character over mistyped text, etc. Additionally, the Nemeth Math code typed may have syntactic errors.

2 Nemeth Math Braille Code

Nemeth Braille Math notation (hereafter referred to as Nemeth code) is a Braille-based notation to enable visually impaired scholars and students to read and write Mathematics [4]. It was designed by Dr. Abraham Nemeth in the late 1940s to allow visually impaired individuals to convey mathematical and technical information. While Nemeth code is well suited for humans, it is not easy to process with a computer. To make Nemeth code computer processable we should be able to *parse* Nemeth code text for grammatical correctness (i.e., check that it conforms to the rules of syntax laid out in the Nemeth code specification [4]), and once parsed, transform it into a form so that it can be processed further. To the best of our knowledge there are no parsers available that will parse Nemeth code to check its grammatical correctness. Two principal reason for this lack of availability of parsers are: (i) context sensitivity, and (ii) spatially arranged mathematical structures.

Context Sensitivity: The Nemeth code notation falls in the class of languages known as *context sensitive languages* [1]. Context sensitivity means that the grammatical correctness of a subexpression depends on the context in which this subexpression occurs within a larger expression. This is similar to the case of English, where, for example, to check the grammatical correctness of the sentence one has to make sure that a singular subject noun must have a singular verb. So `The man eats an apple` is grammatically correct, while `The man eat an apple` is not. Another negative point about the Nemeth code is that it is not an *LALR(1)* [1] language. This means that if we were to write grammar rules for Nemeth code, then given a Nemeth code expression that we are trying to parse, we cannot tell by looking ahead one symbol in the expression which grammar rule should be applied to parse the expression. Thus, a grammar written for a LALR parsing tool such as `YACC` will generate too many *conflicts* [1] at the time of parsing. As a result, traditional parsing and compiling technology [1], that is normally used to process computer languages, cannot be used.

It should be noted that properties of the Nemeth code—especially context sensitivity—are precisely the reason why Nemeth code is so suitable for a visually impaired person. Since a blind person reads by touching the Braille code, it is important that he/she be aware of the context of the current symbol or expression (s)he is currently looking at, at any given time. Thus, building-in the context into the language is quite helpful for blind "readers," but makes its computer processing quite difficult. For example, the expression $x^{z+2^c+3} + y$ is typeset as `$x^{{z+2}^{c}+3}+y$` in LaTeX, since to a sighted reader (or a computer) the braces indicate the scope of exponents. However, this does not work well in a Braille setting, as the reader will quickly forget the exponent level as he/she moves left to right in the formula; considerable number of backtrackings of the finger will be needed to understand it. In Nemeth code, to make the context explicit, this formula is coded as: `x^z+2^^c^+3"+y`. This makes parsing hard, as a context free grammar that can count the number of ^ and check that their number increases or decreases by 1 as we go up and down the exponent level cannot be written [1].

Also, at the time the Nemeth code was designed (late 40s and early 50s), computers were just being invented. Thus, computer readability or computer processability was not one of the design criteria for Nemeth code. Additionally, very little was known about structure of formal languages then. It is very likely, the Nemeth code was designed with the intention that it will be translated into print by humans, who can easily process context sensitivity. Recently, techniques have been developed by Gupta, Guo, and Karshmer—incorporated in the Winsight system—that allow one to build a parser very rapidly and transform Nemeth code into LATEX [3,2].

Spatial Arrangements: Another feature of the Nemeth code that makes it hard to parse are its spatial arrangements of Mathematics [2]. There are mainly two kinds of spatial arrangements in Nemeth code. The first is the spatial arrangements for arithmetic operations such as addition, subtraction, multiplication, and division (including synthetic division), and square-root. The second is the spatial arrangements for matrices, determinants, hypercomplex fractions, and unified expressions. Parsing spatial arrangements is a difficult task.

The backtranslator that is part of the Winsight system parses spatial arrangements by analyzing them several lines at a time to look for patterns within horizontal lines as well as along vertically aligned characters. It also identifies boundaries of non-spatially arranged mathematical expressions that are part of the spatial structure (e.g., a polynomial element inside a matrix). Winsight then translates the non-spatially arranged structures into LATEX. These LATEX translated expressions are then arranged by Winsight into the appropriate spatial structure using LATEX's constructs and then output.

The representation of spatial arithmetic in Braille is guided by numerous rules [4]. These rules have to be used as guidelines in the spatial reasoning module of the Winsight system to accomplish the translation of spatially arranged arithmetic and algebra.

In simple addition, the units digit, tens digit, etc., should always be aligned. Thus, in Nemeth code the addition of two numbers 25 and 125 can be represented only in the following way:

```
  1
77777
  25
+125
33333
 150
```

Notice that 5 (the units digit) in 25 is vertically aligned with 5 in 125 and so on. The operator '+' or '-' should always be present at least one column to the left of the longest element in the operation. The "333333" notation is used as the separator line below which the answer is written. The length of this operator column is at least one greater than the largest row. The carry over numbers are distinguished in Nemeth code by using the notation "7777777". Any number above this row is recognized as a carry.

The restrictions become more complex when dealing with algebraic arrangements. In the following Nemeth code example:

```
  2x^2"-25x+ 7
 25x^2"    +25
33333333333333
 27x^2"-19x+32
```

notice that The `x^2` terms, `x` terms and the constants are vertically aligned. The `+/-` sign are also vertically aligned.

Handling other spatial structures can be likewise complex. For example, there are at least 12 different symbols for spatially representing long division [4]. The division notations in Nemeth is formed using curved division signs made of either '(' or ')' on the right or left, horizontal separation lines which are either in the line above or below, vertical separation lines which are to the right or left, or slanting lines to the right or left. Since any division sign is composed of at least two different types of above mentioned symbols, Winsight has to scan two lines and analyze them before concluding that the input is indeed a division symbol.

In general, Braille to LaTeX translation of spatially arranged arithmetic/algebra cannot be done on a character by character or line by line basis; multiple lines need to be analyzed before conclusions regarding the type of spatial arrangement can be drawn. The same is true for spatial arrangements involving algebraic expressions (e.g., division of a polynomial by another polynomial), these have to be handled by separating algebraic expressions from spatial information, processing them separately, and combining them to create final output. The Winsight system handles all types of spatially arranged structures in Nemeth code.

3 Overview of the Winsight System

The Winsight system backtranslates not just mathematics but complete *mathematical documents*, i.e., documents that contain Mathematics mixed with text. Mathematics is seldom written by itself, it is generally embedded in text. In a Braille document, this text is written in contracted Braille. The Winsight system takes a hard copy Braille document as input, and produces a print file (in pdf) containing the equivalent Math and text. The composition of the Winsight system is shown in Figure 1. Arrows with the label of "M" indicate points at which the user has the option of examining the output produced by that module and changing it.

Winsight OBR Module: The Winsight system takes a JPG file containing image of a hard copy Braille sheet. This image is obtained by scanning the Braille sheet on an ordinary scanner. The scan records the image of shadows of

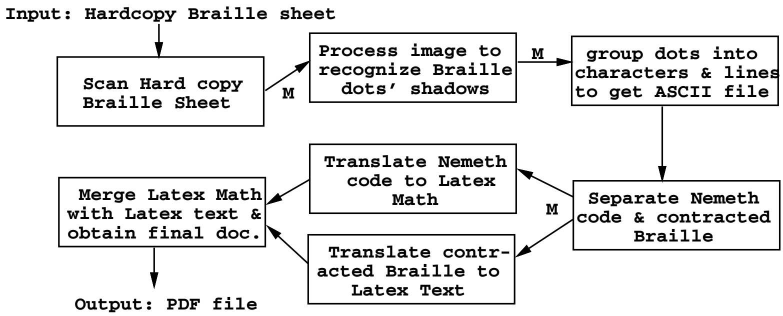

Fig. 1. The Winsight System Flow Diagram

the Braille dots. The user can edit the JPG file using the MS Paint program from within the Winsight system. This editing may be needed to remove smudges on the page or marks left after removing staples. Next, the Winsight system processes the image of the shadows to detect the Braille dots as "blobs."

Next, a sophisticated reasoning algorithm deduces the characters represented by these dots to create an ASCII Braille file. Numerous challenges arise in this step: since the Braille sheet is more likely to be manually produced, the spacing between the lines may be uneven (the lines may be too close), or the paper may be tilted during typing, causing one line of Braille to be slanted w.r.t. another. The paper may have been slightly moved in the middle of the line while typing, causing the remaining part of the line to be slightly misaligned. These problems arise more in hand-typed Braille document; Braille documents produced using an embosser are evenly spaced and do not have such errors. To the best of our knowledge, OBR systems available in the market are designed for documents produced on embossers, and thus are not suitable for hand-typed Braille sheets. In contrast, the Winsight system's OBR module works very well with hand-typed Braille sheets. However, on occasion it will ask the user a few questions (for example, if every Braille character present in the sheet consists of only dots present in the bottom two rows, then the system will ask the user if the top line contains only bottom two rows or top two rows, etc.) and issue warnings (e.g., if two lines are too close). The user can further change the output of the image recognition in two ways:

1. The user can examine the "blob" file and fix any image recognition errors by comparing it with the hard copy Braille sheet. This is helpful when the Braille sheet contains Braille characters that have been erased by back-spacing. In such cases, the erased dots don't completely go away, and their shadow may appear in the scanned image. The user does not need to have any knowledge of Braille for making these fixes.
2. The user can look at the generated ASCII Braille file, and fix its contents. This, however, requires knowledge of mapping between Braille characters and ASCII codes.

Winsight Separation Module: The Winsight system next attempts to automatically separate contracted Braille (Grade II Braille) text from Nemeth Braille Math code. This is a tricky process as it is hard to tell contracted Braille apart from Nemeth code. The same Braille character string can mean different things depending on whether we interpret it as contracted Braille or as Nemeth code [5]. For example, consider the Braille string `y "k t`. If we interpret it as Nemeth code, it will mean $y < t$, while if we interpret it as contracted Braille then it will stand for *you know that.* Likewise, the Braille string `h;e` means h_e in Nemeth code, but *hence* in contracted Braille, and `r+s` means `r+s` in Nemeth code, but *rings* in contracted Braille.

Thus, the general problem of automatically separating Nemeth Braille code from contracted Braille is quite difficult as one has to examine the context. The Winsight system uses a number of heuristics, including knowledge of English words as well as knowledge of various operators and mathematical patterns that can arise in Nemeth code to automatically annotate Nemeth Math expressions in the document. This annotation is done by enclosing Nemeth Math expressions within `;;` and `;'`, as stipulated by Dr. Nemeth [5]. The notation proposed by Dr. Nemeth recognizes a number of subcases, however, Winsight does not support those subcases. This is due to our belief that these subcases are hard to remember for students, and they unnecessarily complicate the notational convention. Future versions of Winsight may support it.

Winsight provides the user the option of viewing the annotated file. A separate option within Winsight allows the user to insert or delete the Nemeth math indicator tags. The math indicator tags, while internally stored as `;;` and `;'`, are displayed to the user as and for ease of visibility and modification. Note that the students themselves can always enclose the Math portion within the `;;` and `;'` tags while typing the Braille document on the Brailler. Expressions within such tags will be treated as Math.

Backtranslation: Once the Nemeth Math code portions and contracted Braille portions of the document have been separated, they are separately backtranslated.

1. The Nemeth code backtranslator in turn then separates mathematical expressions from spatial structures. Thus, given a matrix of of algebraic expressions, the individual algebraic expressions will be isolated and backtranslated separately, the structure of the matrix analyzed separately, and then the two put together to produce a LaTeX rendering of the Matrix. If the Nemeth code backtranslator fails to parse an expression, it is output unchanged. This failure in parsing is usually due to grammatical errors committed during typing the Braille document. The user can fix these errors in the annotated ASCII Braille file and backtranslate again. The Winsight system attempts to fix some errors automatically, but as is well known, automatic error detection and correction during parsing is extremely difficult. Note that our intent is that the backtranslation tool will be used by teachers of blind students. Generally speaking, the Winsight system will be used for grading answers to homework and exams, and in such cases, the teachers have a very good idea

as to what is expected as an answer, and so are likely to be able to fix the typos themselves. They can also seek the students' help in fixing the mistake, as the teachers themselves may not know Braille. Our practical experience indicates that with some rudimentary knowledge of Nemeth code, such typos are relatively easy to fix. To aid in this process, the Winsight system also provides an interface in which a user can type stand-alone ASCII Braille strings, and backtranslate them assuming either that it represents Nemeth code or that it represents contracted Braille. This facility allows a person who does not know the Nemeth code well to try the various fixes that could be possible for the typo made by the student.

2. The contracted Braille backtranslator examines each word, and using the standard mapping of English phonemes and syllables to Braille symbols, backtranslates the whole text. The contracted Braille notation also has context sensitivity and ambiguity built-in and many problems arise during the backtranslation. For example, the Braille character '4' can mean different things in different context. Thus, '4connect' in contracted Braille means *disconnect*, '4t4' means *dist.* (distance or district) while 'ha4ock' means *haddock*. Similarly, the number '0' in contracted Braille can be mapped to *was* or *by* if it occurs by itself. The Winsight system will output *by/was* for a stand-alone 0, relying on the sighted reader to figure out which one was intended.

Final Output: Once the Nemeth Math part and the contracted Braille parts are backtranslated, they are merged to produce the final document in LaTeX. The final LaTeX document is processed and displayed in PDF format, that a sighted user can then read. Note that the Winsight system has been extensively tested on real homework solutions. Below we show part of an example annotated ASCII Braille file produced from a scan (from an actual student homework), and the result of backtranslation.

Annotated ASCII Braille file

```
,homework    ,n ,y>i*
#a4 ;;  ?1/2#+?1/6# .k ?4/6#  ;'
#b4 ;;  ?3/4#+?5/8# .k ?11/8#  ;'
#c4 ;;     160  r2
          33333333333
         4o674
           4
          33333333333
           27
           24
          33333333333
            34
            32
          33333333333
             2 ;'
```

Resulting Print Output

```
Homework   N Yarich
```

1. $\frac{1}{2} + \frac{1}{6} = \frac{4}{6}$
2. $\frac{3}{4} + \frac{5}{8} = \frac{11}{8}$
3.

$$\begin{array}{r} 160\ r2 \\ 4\overline{)674} \\ \underline{4} \\ 27 \\ \underline{24} \\ 34 \\ \underline{32} \\ 2 \end{array}$$

4 Conclusion

In this paper, we gave an overview of the Winsight system for backtranslating complete mathematical documents written using Nemeth code and contracted Braille to print. The system takes as input the scanned image of a Braille sheet, performs image processing to recognize Braille characters, and then backtranslates the resulting ASCII Braille file to LaTeX. The Winsight system is an innovative system, as there is no other system at present that can provide complete backtranslation. Future work includes refining our system to provide better error detection and correction, so as to minimize human intervention.

Acknowledgment. Thanks to Susan Osterhaus of TSBVI, Susan Jolly of Los Alamos, and Dipendra Manocha of the National Association for the Blind, New Delhi, India, for discussion and comments, and to the Department of Education and the National Science foundation for support. Views expressed in the paper are solely those of the authors.

References

1. Aho, A., Ullman, J., Sethi, R.: Principles of Compiler Design. Addison-Wesley, Reading, MA (1986)
2. Annamalai, N., Gopal, D., Gupta, G., Karshmer, A., Guo, H., INSIGHT,: A Comprehensive System for Translating Braille based Mathematical Documents to LaTeX. In: Proc. HCI'03. pp. 1245–1249 (2003)
3. Karshmer, A., Gupta, G., Geiger, S., Weaver, C.: A Framework for Translation of Braille Nemeth Math to Latex. In: Proc. ACM ASSETS'98, pp. 136–143 (1998)
4. Nemeth, A.: The Nemeth Braille Code For Mathematics And Science Notation 1972 Revision. American Printing House For The Blind (1972)
5. Nemeth, A.: The Nemeth Uniform Braille System (NUBS). Draft (April 2003) `http://www.braille2000.com/brl2000/docs/NUBSdraft1.pdf`
6. Scadden, L.: Making Mathematics and Science Accessible to Blind Students Through Technology. In: Proceedings of RESNA'96 (1996)
7. Lamport, L.: LaTeX A Document Preparation System. Addison Wesley, Reading, MA (1994)
8. `http://www.mackichan.com`