TALKMATHS OVER THE WEB : A WEB-BASED SPEECH INTERFACE TO ASSIST DISABLED PEOPLE WITH MATHEMATICS

Dilaksha Attanayake School of Computing and Information Systems, Kingston University, UK
James Denholm-Price School of Mathematics, Kingston University, UK
Gordon Hunter School of Mathematics, Kingston University, UK
Eckhard Pfluegel School of Computing and Information Systems, Kingston University, UK

Abstract : Difficulties in accessibility to resources for, and writing documents in, mathematical notation, has limited the educational and career opportunities of people with disabilities such as visual impairments and/or limited (or no) use of their hands or arms. In this paper, we describe a system, *TalkMaths*, aiming to help address these issues. *TalkMaths* is a speech interface allowing users, particularly those with the type of disabilities noted, to dictate and edit mathematical text, using relatively simple natural language commands. Although a PC-based version of *TalkMaths* was developed a few years ago, the latest version uses a web-based client-server architecture, which reduces the demands on the user's own computer, and potentially making the system usable on mobile devices. We discuss the technology – including automatic speech recognition, statistical language models for prediction and correction, and novel parsing strategies - underpinning *TalkMaths,* and present results from some initial evaluations of its use both as a dictation/editing system and as an instructive tool.

1 INTRODUCTION

Information and Communication Technology (ICT) has made a very great impact on education, in the classroom and elsewhere, over the last few decades. For example, it is now common for teachers to use "smartboards" and similar facilities to annotate notes in front of their class, then save the resulting "hybrid" document for their students to download and review in their own time. This greatly reduces the burden on students for taking accurate notes at high speed during classes, which was considered normal 20 years ago. In those days, it often proved a challenge to students to write down all the relevant information correctly whilst at the same time listening to the tutor. Now, students should be able to focus on reading the material, listening to what the teacher is saying, and trying to understand the concepts being taught.

Mathematics is a core subject of study in the school curriculum in most countries, and proficiency in it, at least at an elementary level, is essential for success in a wide range of scientific, technical and commercial fields. However, in many ways, Mathematics is perhaps less well-suited to be taught with the aid of modern educational ICT than many other disciplines. It is a subject which many students find difficult, partly due to its specialized language and notation. These make working with mathematical equations and formulae a problem for a large proportion of people. This is an even bigger problem when the mathematical expressions to be manipulated are to be included in electronic documents. Typing and editing ordinary text can be both slow and error-prone for non-experts, and this is even more the case for mathematical text, with its non-alphanumeric symbols and typically somewhat complicated two-dimensional layout. Furthermore, creating, editing and reading mathematical text (in its conventional form) is particularly difficult for three types of groups of people : individuals suffering from various disabilities[31, 32], people relying heavily on on-line learning systems[9] (particularly distance-learning students), and people relying on using portable devices, such as smartphones and tablet computers[9], to access learning resources. These factors can severely limit the educational and career opportunities of such groups.

In this paper, we first review some existing recent approaches to addressing these problems. We then focus on our own approaches, implemented via our system *TalkMaths*, namely using spoken input (processed by an automatic speech recognizer, then parsed and analysed) as an option for creating and editing mathematical text in electronic documents, and providing an "intelligent assistant" to aid the user by predicting what may come next and offer a semi-automatic correction facility to rectify mistakes. We describe how our system has been designed and implemented, discuss how it has been evaluated as a possible educational tool, and describe current work in progress and suggest potential future developments to it, with a view to making it a powerful and flexible tool which could be of benefit to a wide variety of people, both with and without disabilities.

2 EXISTING ASSISTIVE SYSTEMS FOR MATHEMATICS

As noted above, typing and editing mathematical text poses particular problems for people with disabilities – notably visual impairments and/or loss of (or the use of) hands or arms. Various systems, either research prototypes or, in some cases, fully developed tools, have been created to help such people with these tasks. We first discuss some previous existing assistive technologies, developed by other groups, before describing the original, desktop computer-based, version of our *TalkMaths* system. In section 3, we then discuss how we have recently improved *TalkMaths* by transforming it into a web-based service.

2.1 Other (Previous) Systems

Various approaches to address the problems faced by disabled people trying to use mathematical notation have been proposed and implemented in the past. These have been previously reviewed in some detail previously[4,5]. In summary, each such solution has tended to focus on alleviating the difficulties of one particular group of users – for example, to provide Braille (extended to support mathematical notation) input or output for the blind and severely visually impaired[20]. Similarly, for certain groups of people – including those with certain types of disabilities, such as repetitive strain injuries[31, 32], those using small mobile devices and some types of on-line distance learners[9] - typing, but not writing, mathematics would be a problem. For such individuals, a possible modality for input of mathematical expressions might be to write the formulae and equations using a smart pen or stylus. Optical character recognition software could them be used to identify the mathematical symbols used, which would then be converted into (correctly) typeset mathematical text in electronic form – an approach adopted by the INFTY project[28]. Other groups, including the visually impaired, people with limited use of their hands or arms, and people using portable devices, could benefit by the input and/or output modalities being through speech. There have been a variety of systems which attempt to provide synthetic speech descriptions of mathematical text[12, 13, 16, 24, 27], and others allowing spoken input of mathematics[7, 14, 15, 22]. However, all of these systems have serious limitations, prompting us to develop our own system, *TalkMaths*.

2.2 Previous Version of *TalkMaths*

The original, desktop PC-based version of *TalkMaths* had been under development since 2007, and has already been the subject of several published articles[2, 3, 4, 30, 31]. In summary, *TalkMaths* version 1.0 accepted spoken descriptions of mathematical expressions, in relatively natural language, as input. These were first transcribed, using a commercial automatic speech recognition system, parsed using a LL parser[1, 18] with a bespoke Context Free Grammar (CFG), converted to a description in a mathematical mark-up language (either *LaTeX* or *MathML*), then rendered into a form suitable for display on a VDU or similar screen in conventional mathematical notation[31, 32].

The choice of type of spoken input language for describing mathematical expressions required something of a compromise, between a precise, highly prescriptive (but possibly rather difficult to learn) language such as those advocated by Chang[8] and Rahman[24], and a more intuitive, easy to learn but imprecise and potentially ambiguous type of mathematical language of the type spontaneously used by students and teachers to “read” mathematical expressions aloud in the classroom[32]. In the end, we tried to adopt a “middle way”, rather like the approach used by Fateman[11], making the language sufficiently formal to avoid ambiguities whilst attempting to make it close enough to the way mathematically proficient people speak or read mathematical equations and formulae, in order to make the language relatively easy to learn and use[5, 32].

The desktop-based design/architecture of this system[31, 32] had certain advantages, as it could be used off-line. However, it had quite heavy requirements on local resources (memory and computational processing power) and also required quite a sophisticated local Graphical User

Interface (GUI), which therefore required quite a complicated local GUI program and rather intensive local processing of input. With these issues in mind, we decided to investigate the possibility of a web-based implementation of *TalkMaths*, which is the focus of the following section.

3 WEB-BASED VERSION OF *TALKMATHS*

The current version of the TalkMaths system uses a web-based architecture, as described by Attanayake et al[2, 5]. The application logic, that controls the behaviour of the application independent from any speech input, in form of an interactive web application, is hosted on an external server and the main parsing logic, the Post-ASR Processing (PASRP) - which deals with interpreting the input spoken commands once they have been transcribed - is hosted on a separate server. Both these servers may be remote from the client PC, but connected to it via the Internet. Speech recognition is executed on the user's local machine, by an ASR system of choice (currently, *Dragon NaturallySpeaking*[TM] returns the best recognition accuracy) with the additional support of a speech-front-end for customising the vocabulary and providing some basic support for the spoken commands. The actual interface for entering commands is done through an input field on the web interface, but this can actually be evoked by typing commands as well.

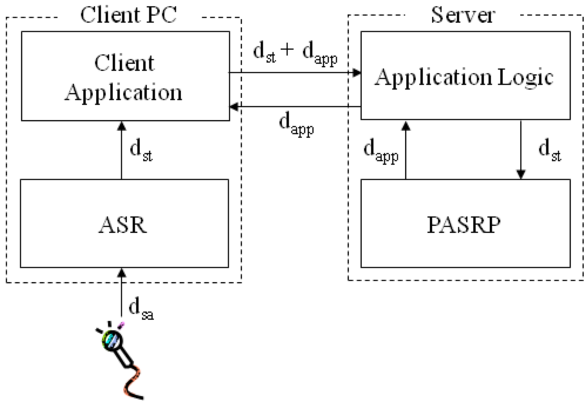

Figure 1 : The system architecture of the newer, web-based version of *TalkMaths*[2,5]. The automatic speech recogniser (ASR) and the Client Application (e.g. for display of the output on a screen) run on the local "Client" PC, whereas the Application Logic and interpretation of the commands takes place on Server, which may b remote from the client PC. PASRP stands for Post-ASR Processing, which carries out the interpretation of the transcribed spoken commands. d_{sa} stands for speech audio data, d_{st} for transcribed speech data, and d_{app} for application-specific data.

Various other system architectures were considered[2,5], but the advantage of this architecture adopted is that users can access the system with a web browser, provided they have local speech recognition and the speech front-end installed. Whilst, in principle, use of any standard web browser would be suitable, at the moment, the recommendation is to use Google Chrome for best rendering performance.

The parsing engine is based on that of Wigmore[31, 32], but with a slightly enhanced grammar. After creating a parse tree, this is converted to XML and finally XHTML for display in the browser. Additional information, such as coloured boxes and labels, displayed with the support of Cascading Stylesheets (CSS), is used for speech-navigating nested expressions in the context of editing.

4 EVALUATION OF TALKMATHS AS AN EDUCATIONAL TOOL

Issues relating to on-line and distance learning of mathematics have been investigated and speculated upon for many years. Before the inception of the *TalkMaths* project (circa 2006 – see Wigmore *et al*[30,31]), Smith & Ferguson's 2004 study[26] into then state-of-the-art of mathematics communication tools concluded that "most online mathematics instructors still wait for a simple and convenient way to communicate two-way with their students in the very language of mathematics". Whilst the sophistication of tools has increased, from *MSN Messenger* in the study of Loch and McDonald[19] through to online "whiteboard" and screen-sharing tools, such as "*Elluminate Live*!" that was evaluated at the Open University[21], such "synchronous communication tools" still present barriers to some disabled users, as they function most effectively with pen-like input devices (which are not suitable for people with no or limited use of their hands) and present barriers to users of screen-reading software, such as the visually impaired. Synchronous tools also tend to be evaluated in the context of supporting existing pedagogies, but the different modalities they encourage may suit different ways or styles of learning.

We tested the potential utility of TalkMaths as an educational tool in a small-scale evaluation on students learning mathematics within another main discipline (Life Sciences) – see Attanayake *et al*[4,5] for full details. For those students, the level of mathematical notation that *TalkMaths* supported was sufficient to exceed their current knowledge and the experiment compared the recall and understanding of some new mathematical concepts between two groups of students who either used *Microsoft Word® Equation Editor* or *TalkMaths* during a tutorial exercise (students were assigned to one or other of the two groups randomly). The results suggest that, compared to using *Equation Editor, use of TalkMaths* made a small, but statistically significant, improvement to their understanding of the mathematical concepts being tested, which also matched their self-evaluation of their performance. Very few of the participants found *TalkMaths* to be "easy" or "very easy" to use compared to the majority of users of *Equation Editor*, which is not surprising given that all of the participants had used *Equation Editor* before, whereas none of them had previously used *TalkMaths*. This admittedly small, pilot study gave encouraging results and suggests that using the web-based version of *TalkMaths* with a remote collaboration tool such as *Elluminate* or *Blackboard Collaborate* could be beneficial to distance learning of mathematics and be accessible to a wider range of users than the collaboration tool alone.

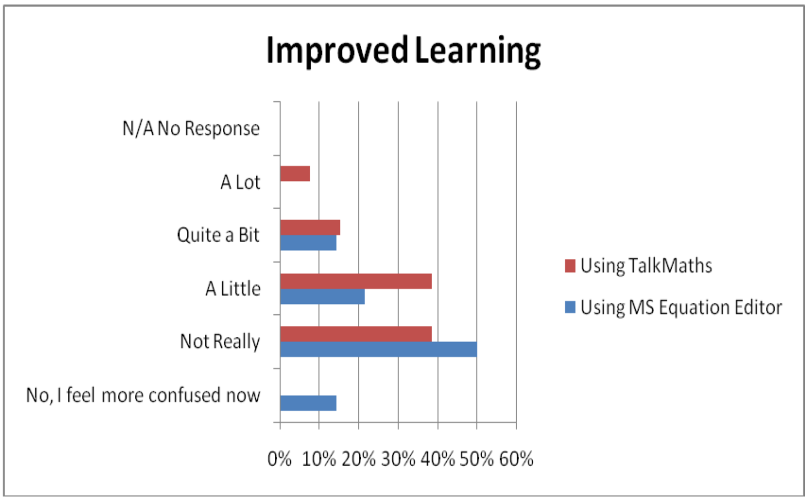

Figure 2: Students' perceptions of whether, and by how much, using *TalkMaths* or *MS Word Equation Editor* in a tutorial exercise improved their understanding of the mathematical concepts involved. It can be seen that students were more positive about TalkMaths helping their learning than they were about Equation Editor. The full details of this study are given in Attanayake et al[4, 5].

5 CURRENT WORK IN PROGRESS – IMPROVING *TALKMATHS*

5.1 Use of Statistical Language Models for Prediction and Correction

A wide variety of text dictation and editing systems employ prediction and/or correction methodologies in an attempt to make the systems more useful and usable. These include automatic (or semi-automatic) correction systems found in word processors and internet search engines ("Showing results for … Search instead for …") and the prediction systems used in Automatic

Speech Recognition (ASR) systems and SMS text message editors on mobile telephones. Many of such prediction systems use statistical language models (SLMs)[25], whereas correction systems often look for "close matches" to what was entered from a database of common words or phrases. These statistical models give probabilities of words and word sequences, using information from a large corpus of previously observed data and evidence from the current context together with an "inference rule", such as a Bayesian framework, in order to combine information from more than one source[33]. The simplest types of SLMs are N-gram models[33], which use statistics of the occurrences of specific sequences of N consecutive words within a database (or "corpus") of training material observed in the past. Individual words (N=1) are referred-to as unigrams, pairs of consecutive words (N=2) as bigrams and triplets of consecutive words (N=3) as trigrams. Regarding the correction of mistakes, it has been noted that the majority of human typing and spelling errors are quite minor, often involving just the omission or addition of a single character, typing two characters in the wrong order, or accidentally substituting one character for another (often one adjacent to the correct symbol on the keyboard). The Damerau–Levenshtein distance[10] measures how different two strings are by calculating the minimum numbers of insertions, deletions, substitutions and transpositions of characters required to transform one of the strings into the other.

In attempt to incorporate some functionality to assist the user by trying to predict what (spoken descriptions of which) mathematical symbols likely to be input next, we investigated the mathematical content of a number of mathematics tutorial websites, covering topics at "upper high school level", in the public domain. From these, we obtained a total of 4100 mathematical expressions (equations, formulae, etc.), which we then converted into our spoken mathematics description language. This yielded a total of 77 824 words, with a vocabulary of 100 words. From these we constructed unigram, bigram and trigram models[3,5]. These models were then applied in a set of cross-validation experiments in order to predict the next word or two words to appear in the spoken mathematics description of an expression. Dependent on the number of possible choices offered to the user, correct single word prediction (given the two previous words) rates of up to about 95% could be achieved, but for prediction of the next two words (based on only one previous word), only success rates close to but below 45% could be attained[3, 5]. These results illustrate that incorporation of some predictive feature into *TalkMaths* could potentially assist the user, but the benefit of using our current SLMs and prediction method is likely to be limited to predicting the next word. Further research is being carried out in this area.

We also studied the possibility of using an error correction process, based on the Damerau–Levenshtein metric, to identify possible alternatives to words not in our mathematical vocabulary. For any such "out of vocabulary" word found, our system would identify the closest matches, namely those with the smallest Damerau–Levenshtein distance, amongst words which were in the mathematical vocabulary[3,5]. We tested this approach by artificially introducing a small number of errors into otherwise correct spoken mathematics descriptions of expressions in our dataset. It was found that this method could successfully correct 97% of examples from which three characters had been deleted, and 91% of cases where three pairs of adjacent characters had been swapped. The method was less successful in correcting examples with several deleted characters – only 68% of cases with three characters deleted were successfully corrected, but 92% of examples from which just two characters had been removed[3, 5]. These show that this methodology could be of use for correcting words which had been mis-recognised by the ASR system, which could then cause TalkMaths to fail to interpret the input as the user intended. It is proposed to integrate both these prediction and correction systems into TalkMaths in the near future.

5.2 Novel Parsing Methods – Generalised Operator Precedence Parser

The previous version of our system exhibited a number of disadvantages as far as the parsing of the input was concerned; mainly due to the use of the open source parser generator tool yapps2 ("Yet Another Python Parser System", version 2"[23] as the means of analysing the transcribed spoken mathematical input. Using yapps2 meant employing a LL-based parsing strategy using a recursive descent process[1, 18]. We noted that, in practice, people might wish to preview expressions which they had not completed dictating, or to get feedback from the system if they had made an error or given ambiguous input which could potentially be interpreted in several ways. Hence, spoken input may sometimes be incomplete or contain ambiguities or errors. The LL approach was less flexible when trying to extend the grammar in order to cope with such incomplete input or allow error recovery. Also, it was noted that the CFG, as specified by Wigmore[32] contained some mistakes which led to the wrong interpretation of certain complex expressions, but trying to correct these mistakes seemed to become complicated as the number of grammar rules increased rapidly in order to allow more flexibility and deal with more advanced mathematical notation. The speed of this parser was also an issue, as recursion, on which the LL approach is based, tends to be slower than iterative approaches, such as table-driven LR parsing[17].

Our ongoing work is concerned with a new parsing mechanism, which at the same time is closely related to our design of the actual spoken input language, as described in section 2.2 above. Our starting point is a design framework using *mixfix operators*, leading to the construction of commands that we refer to as *speech templates*[5]. (These should not be confused with the type of acoustic templates used in some early speech recognition systems.) Mixfix operators are generalised operators that can consist of several operator parts and can take more than two operands. This leads to a collection of short and reasonably intuitive spoken commands, that can be nested arbitrarily, parsed easily and also allow for partial and/or incomplete input, as noted above, to be processed[5].

The parsing method for our proposed language is based on Operator Precedence Parsing (which is a sub-class of LR-parsing) and non-deterministic parsing techniques[1]. In contrast to the recursive descent context-free grammar parser generator yapps2 that was used in approach is inspired by work on spoken programming languages[6] and our main novel contribution here is an error recovery method for an adaptation of the XGLR parsing approach[6] to our operator precedence setting. As a consequence, the range of mathematics that can be parsed by TalkMaths has been significantly enhanced and, for the first time, our system can tolerate errors that might typically arise from the input of spoken or typed mathematics[5].

5.3 Generalisation to other Domains – *TalkCode* and *TalkDocs*

Our approach can significantly enhance the ease of design and maintenance of any general structured spoken command language, not only our proposed language for spoken mathematics. In particular, spoken dictation of computer program code or other types of structured documents (such as those encoded using document mark-up languages) could be useful application areas. Potentially, this could target an audience of substantial significance as, typically, proficient and prolific programmers are more prone to suffer from upper limb disorders such as Repetitive Strain Injury (RSI) than are other members of the general population.

Two prototype applications, *TalkCode* and *TalkDocs* are currently being developed as new initiatives, based on the existing parsing engine, but extending the approach to use an incremental parser[29]. It is hoped that speech templates, as described above, will, as in the case of spoken mathematics, be a suitable framework for expressing spoken commands for inserting or modifying typical computer programming constructions such as loops, conditional statements and variable declarations.

6 CONCLUSIONS & FUTURE WORK

In this paper, we have described how we have enhanced the previous *TalkMaths* system, which allowed users to dictate mathematical equations and formulae into electronic documents, making it a web-based application. Through a small scale evaluation of this with real students, we have shown that it may have potential as an educational tool to assist the teaching and learning of mathematical concepts. We have described current work in progress to make *TalkMaths* even more flexible, powerful and user-friendly by incorporating improved parsing methods, text prediction via statistical language models and error correction using a string similarity metric.

We propose to further improve *TalkMaths* by incorporating the additional features described above, and also include a user-friendly interface for incorporating new commands and keywords into the vocabulary and/or new operators and rules into the grammar. In these ways, it should be possible to increase the range of mathematical notation with which the system can be used, and improve the ways in which text can be edited and corrected. We are also investigating alternative modalities for input to the system, and how it outputs the interpreted mathematical expressions to the user. These could include Braille input and output and audio output (by synthesised speech) to assist blind or severely visually-impaired users. We also intend to carry out more extensive user trials on the system, with both disabled and non-disabled participants, in order to assess its utility and usability.

7 REFERENCES

1. A.V. Aho, R. Sethi and J.D. Ullman (1986) Compilers : Principles, Techniques and Tools, Addison- Wesley Publishing, Reading, Massachusetts, USA.
2. D. Attanayake, E. Pfluegel, J. Denholm-Price, G. Hunter (2011) Architectures for Speech-Based Web Applications, Proceedings of 4th International Conference on Semantic E-business & Enterprise Computing (SEEC2011), U.K., July 2011, pp 73 - 77
3. D.R. Attanayake, G. J. A. Hunter, J. C. W. Denholm-Price, E. Pfluegel (2012) *SWIMS* (Speech-based Web Interface for Mathematics using Statistical language models): An intelligent editing assistant for mathematical text, Proceedings of 8th Int. Conference on Intelligent Environments, IE2012, Guanajuato, Mexico.
4. D.R. Attanayake, G.J.A. Hunter, E. Pfluegel, and J.C.W. Denholm-Price (2013) Novel multi-modal tools to enhance disabled and distance learners' experience of mathematics, International Journal on Advances in ICT for Emerging Regions (ICTer), 6(1) .
5. Dilaksha Attanayake (2014) Statistical Language Modelling and Novel Parsing Techniques for Enhanced Creation and Editing of Mathematical E-Content Using Spoken Input, PhD Thesis, Kingston University, London, UK.
6. Andrew Begel (2005) Spoken Language Support for Software Development. PhD thesis, University of California, Berkeley, USA, Report UCB-EECS-2006-8.
7. C. Bernareggi & V. Brigatti (2008) Writing mathematics by speech: A case study for visually impaired, Proceedings of 11th ICCHP 2008, pp. 879–882.
8. L. A. Chang (1983). Handbook for spoken mathematics (*Larry's Speakeasy*). Lawrence Livermore National Laboratory, University of California, USA.
9. J. Cuartero-Olivera et al (2012), Reading and writing mathematical notation in e-learning environments, eLC Research Paper Series, Universitat Oberta de Catalunya, Spain, http://elcrps.uoc.edu/ojs/index.php/elcrps/about
10. F. J. Damerau (1964) A technique for computer detection and correction of spelling errors, Communictions of the ACM, 7(3) 171-176, March 1964. ISSN 0001-0782.
11. R. Fateman (2009) How can we speak math ? http://www.eecs.berkeley.edu/~fateman/papers/speakmath.pdf , Accessed: 02/08/2014.
12. H. Ferreira, D. Freitas (2004) Enhancing the Accessibility of Mathematics for Blind People: The AudioMath Project, K. Miesenberger et al (eds.) Proceedings of 9th International Conference on Computers Helping People (ICCHP 2004), Springer LNCS, vol. 3118, pp. 678–685.
13. P. Gaura (2002) *REMathEx* - Reader and Editor of the Mathematical Expressions for Blind Students, Proceedings of 8th ICCHP 2002, Springer-Verlag, London, U.K., pp. 486-493.
14. C. Guy, M. Jurka, S. Stanek, and R. Fateman (2004) *Math Speak & Write*, a computer program to read and hear mathematical input. Technical report, University of California, Berkeley, Electrical Engineering and Computer Sciences Department.

15. T. Hanakovic & M. Nagy (2006) Speech recognition helps visually impaired people writing mathematical formulas, Proceedings of 10th International Conference on Computers Helping People (ICCHP 2006), pp. 1231–1234, Springer, Berlin.
16. N. Jacobs (2006) *MathGenie* : User's Guide and Teacher's Manual, http://logicalsoft.net/MathGenie.pdf
17. D. E. Knuth (1965) On the translation of languages from left to right. Information and control, 8(6): 607-639.
18. P. M. Lewis and R. E. Stearns (1968) Syntax-directed transduction. Journal of the ACM (JACM), 15(3): 465-488.
19. B. Loch & C. McDonald (2007). Synchronous Chat and Electronic Ink for Distance Support in Mathematics, *Innovate*: Journal of Online Education, 3(3).
20. Math in Braille Project, 2011, http://www.mathinbraille.at/en
21. B. Mestel, G. Williams, T. Lowe and G. Arrowsmith (2011) Teaching Mathematics with Online Tutorials, MSOR Connections 11(1), 12-17.
22. Metroplex Voice Computing Inc., *Mathtalk*. http://www.mathtalk.com/ Accessed: 02/08/2014.
23. A. Patel (2007) 'Parsing with Yapps', http://theory.stanford.edu/~amitp/yapps/
24. T. V. Raman (1998) *ASTeR* : Audio System for Technical Readings, Springer, Berlin.
25. R. Rosenfeld (2000). Two decades of statistical language modeling: Where do we go from here ? Proceedings of the IEEE, volume 88(8), 1270-1278.
26. G.G. Smith and D. Ferguson (2004) Diagrams and math notation in e-learning: Growing pains of a new generation, International Journal of Mathematics Education in Science and Technology 35 (5): 681-695.
27. N. Soiffer (2005) *MathPlayer*: web-based math accessibility. Proc. 7th Int. ACM SIGACCESS Conf. on Computers and Accessibility (Assets '05), pp. 204-205.
28. M. Suzuki et al (2003) INFTY – An integrated OCR system for Mathematical Documents, Proceedings of ACM DocEng'03, Grenoble, France
29. T.A. Wagner and S.L. Graham (1998) Efficient and flexible incremental parsing, ACM Transactions on Programming Languages and Systems (TOPLAS), 20(5), 980-1013.
30. A.M. Wigmore, G.J.A. Hunter, E. Pfluegel, S. Jalan and J.C.W. Denholm-Price (2008) "Talk to the Machine : it's listening - but the hand ain't typing !" The Development of the "*KU-Talk*" and "*TalkMaths*" Speech Interfaces at Kingston University, Proceedings of the Institute of Acoustics (U.K.), 30(2), 654 - 661.
31. A. Wigmore, G. Hunter, E. Pfluegel, J. Denholm-Price, M. Colbert (2010) "*TalkMaths* Better !" Evaluating and improving an intelligent interface for creating and editing mathematical text. Proceedings of 6th International Conference on Intelligent Environments, Kuala Lumpur, Malaysia
32. A. M. Wigmore (2011) Speech-Based Creation and Editing of Mathematical Content, PhD Thesis, Kingston University, U.K.
33. S. Young (1996) A Review of Large Vocabulary Continuous Speech Recognition. IEEE Signal Processing Magazine, 13(5) 45 - 57