

# VOICE RECOGNITION TECHNOLOGY FOR SPREADSHEET NAVIGATION AND INFORMATION RETRIEVAL

**Abstract.** Due to the large volume of data that spreadsheets can store, finding meaningful information can be a difficult task. Existing tools can be quite complex and difficult to master, especially for novice spreadsheet users. In the past Natural Language Interfaces have been used to improve the efficiency of using similar tools. This work aims to look at the use of a Natural Language Interface to provide an easier means of interaction with information retrieval tools for novice spreadsheet users.

**Keywords:** Natural language Processing, Information retrieval, Spreadsheet Debugging

## 1 Introduction

When presented with a large volume of data, Information retrieval, which is the extracting of additional information from such a spreadsheet, can be a difficult task. Spreadsheet applications are commonly used for this purpose as they provide features, such as PivotTables, to assist users. These features however can be difficult to operate, especially for novice spreadsheet users. Using these tools requires a large number of interactions, which for users of voice recognition technology can be a time consuming process. Through the use of a Natural Language Interface (NLI), the need to interact with these tools is replaced with a more user friendly alternative.

Consider a spreadsheet containing a list of golf courses situated in the state of Indiana. This spreadsheet could contain information such as the number of holes on the course, the type of terrain and the county in which the golf course is situated. If a user wanted to know how many courses were in each county, this user would need to first select some of the data, then click on the pivot table button. The user would then need to select the location where the pivot table is to be placed either a new worksheet or on the existing one. A blank pivot table would then be generated and the user would need to select the column heading 'County' and assign it to a row or column. They would then need to count on some attribute to determine how many courses are in each county. With a natural language interface a user could simply ask "How many golf courses are in each county?" and the system would return the number of courses that are in each county.

The aim of this research is to investigate and improve the usability and effectiveness of spreadsheet navigation and information retrieval using voice recognition technology. It is believed that by simplifying the way in which users interact with the spreadsheet application their efficiency can be improved. Section 2 of this paper outlines the background to this research.

There are many ways in which voice recognition technology can be used, from controlling portable devices to operating computer applications such as Microsoft(c) Excel. Section 3 outlines some of these technologies and how they have changed since their first inception.

The remainder of this paper details the work that has been conducted into improving the operation of spreadsheets through voice recognition technology. The aims and objectives of this work are outlined in Section 4. Section 5 then outlines an investigation that was conducted into the use of existing voice recognition technology for spreadsheet debugging. This investigation has highlighted a number of issues with this technology and also a possible means of improving it.

One of these is through an alternative navigation system, *iVoice*, outlined in Section 6 which has been developed using lessons learned during the initial investigation. This system provides simple voice commands for complex navigational activities such as moving to a referenced cell on a different worksheet.

Section 7 outlines a proposed NLI for spreadsheet information retrieval. This system will allow users to tell a computer what they want in their own, natural way, eliminating the need to use the complex interfaces that are found with common information retrieval tools.

## 2 Background

Spreadsheets are a ubiquitous software tool. Some organizations, such as those in the financial sector, rely heavily on spreadsheets for implementing complex models such as cash flow statements. Within the financial district of London, spreadsheets have been described as "the primary frontline tool of analysis"[Croll, 2005]. With such complexity, these models often contain errors which can result in substantial financial losses. One such example is the Columbia Housing Authority who overpaid landlords by \$118,387 as a result of an erroneous spreadsheet[Miller, 2006]. Two independent studies [Panko, 1998; Powell, Baker et al., 2007] have shown that over 90% of the operational spreadsheets reviewed contained errors. One way of eliminating these errors is through spreadsheet debugging.

Spreadsheet debugging is a process whereby an individual reviews a spreadsheet for errors and removes those that are found. Such processes are required by regulations, such as the Sarbanes-Oxley Act in the United States, to ensure the accuracy of the accounts being reported. For users of voice recognition technology however this is a difficult and time consuming task[Flood and McDaid, 2007]. Existing voice recognition software only provides limited functionality for spreadsheet applications. Often to complete relatively simple tasks such as moving to another worksheet users are required to use multiple voice commands which can have a substantial impact on their performance.

Spreadsheets can also be used to store large amounts of data such as inventory logs. With such a large volume of data extracting meaningful information can be a difficult task. Inbuilt features such as filtering and PivotTables can assist in this task however for users of voice recognition technology these features can be difficult to operate, especially for those inexperienced with spreadsheet applications.

In the past, NLI's have been used to simplify the operation of computer applications. This can be done in many ways such as reducing the number of interactions required or eliminating the need for complex structured commands. Generating charts[Kato, Matsushita et al., 2002], Querying databases[Zelle and Mooney R. J., 1996] and searching XML files[Woodley, Tannier et al., 2006] are just some of the tasks that have been simplified through the use of NLI's.

### **3 Voice Recognition Technology**

Voice recognition is nothing new, since the 1960's people have talked about talking to computers, however at the time computer power was very limited and as such voice recognition was extremely limited and time consuming.

Since that time however, computer power has increased exponentially and voice recognition has come a long way. Early systems relied on a limited vocabulary meaning systems could only recognise certain key words. These early systems, known as Discrete-word Recognition systems[DeHaemer, Wright et al., 1994], required users to pause between each word they said so that the voice recognition could decipher when one word ends and the next begins.

Today, through the use of continuous voice recognition, interaction between a user and their computer has become more natural. No longer do users need to pause between each word they say, instead they can speak at their own natural pace secure in the knowledge that the voice recognition engine will be able to decipher what they are saying.

There are a number of voice recognition systems available that cater for a wide range of applications from word processing to in-car control of mobile devices. "Sync: voice activated In-car technology" allows users to connect mobile devices, such as a mobile phone or an MP3 player to their car, and to operate these devices with their voice.

There are many voice recognition applications suitable for the home PC such as Dragon NaturallySpeaking (DrNS). There are currently 5 versions of this popular dictation software on the market. The most basic of these, Standard edition, features support for Microsoft® Word and web browsing through Microsoft® Internet Explorer. The next edition, the preferred edition, includes support for Microsoft® Excel. The professional Edition builds on this again, by offering support for all Microsoft® Office applications. There are also two special editions, Legal edition and Medical edition, which provide recognition for specialised legal and medical terms (<http://www.nuance.com/naturallyspeaking>).

An alternative to DrNS is the Microsoft Speech API. It has two modes of operation, "command and control" and "dictation". The "command and control" mode allows users to use predefined commands. While in this mode the software will only recognise words that have been defined as commands. While in "dictation" mode the software will interpret words and phrases as text and will ignore any commands that may be associated with these words. Users are required to manually switch between the two modes, which is often a source of frustration for new users. DrNS on the other hand integrates both modes.

### **4 Research Objectives**

The aim of this research is to investigate and improve the usability and effectiveness of voice recognition technology for two common spreadsheet tasks: Spreadsheet debugging and information retrieval.

The research first looked at spreadsheet debugging. The work recognized that any improvements should be evaluated relative to the current standard of voice recognition technology for this task. This standard was examined through the following two research questions:

- **RQ1:** How does the performance of those who debug spreadsheets using voice recognition technology compare to the performance of those using a keyboard and mouse?

- **RQ2:** What are the issues associated with debugging spreadsheets through voice recognition technology?

The investigation into voice recognition technology, outlined below, revealed that navigation was a critical factor in spreadsheet debugging. This was found while looking at the frequency of navigational activity within this investigation and on the time it took to perform these navigational activities. Based upon this information, a further research question was posed to identify ways in which voice controlled navigation could be improved.

- **RQ3:** Can navigation behaviours be observed that could be utilized to improve spreadsheet debugging through voice recognition technology?

An intelligent navigation system, termed *iVoice*, was built to replicate some of the behaviours that were observed while answering RQ3. When the system was complete an additional research question was posed to investigate whether or not the system improved the performance of those debugging spreadsheets using voice recognition technology.

- **RQ4:** Can the performance of voice recognition users debugging spreadsheets be improved with an alternative navigation system?

The second aspect of our work examines information retrieval from spreadsheets. This task requires multiple interactions which can be time consuming. Research in other domains has demonstrated that such interfaces can be replaced, with greater efficiency, with Natural Language Interfaces. It is our belief that similar benefits will be seen through a NLI for information retrieval.

Before developing such an interface, the type of language that would be used needs to be investigated. This investigation will allow for the development of a suitable grammar on which to base the interface.

- **RQ5:** How would users phrase information requests to a natural language interface for spreadsheets when given a tabular structured spreadsheet?

Once the grammar has been established, the natural language interface could be developed to give the users the information they require. An additional research question is used to investigate the effectiveness of such a system:

- **RQ6:** Can a natural language interface be used to improve information retrieval for novice spreadsheet users?

## 5 Investigative study

In an experiment described in [Flood and McDaid, 2007] the authors found that spreadsheet auditing through voice recognition was slower and less accurate than traditional keyboard & mouse control.

The experiment asked three experienced spreadsheet users to audit a spreadsheet seeded with errors using a state of the art voice recognition technology, DrNS. For the experiment, the Preferred edition of DrNS was used as it facilitates the use of voice recognition within Microsoft© Excel, which was used as the spreadsheet application. DrNS preferred edition also contain additional commands to simplify some common spreadsheet activities, such as adding up columns. From the authors experience, it was found that DrNS was the most efficient when compared with Microsoft's© in-built voice recognition engine. None of the participants had any prior experience with voice recognition, however they had sufficient experience with spreadsheets to complete the task easily. Their performance was compared with another study[Bishop and McDaid, 2007] in which Bishop asked 13 professional spreadsheet users to audit the same spreadsheet using a keyboard and mouse.

The study recorded the behaviour of each participant through logging of cell selection activity via the T-CAT[Bishop and McDaid, 2008] (Time-stamped Cell Activity Tracker) tool. This technology was also used to collect the cell selection activity of the voice-controlled subjects. Overall it was found that, despite taking twice as long as the keyboard and mouse participants, those using voice recognition found 14% less errors.

Prior to the experiment a number of tasks were identified as potential problem areas for voice recognition users. These tasks were as follows: Navigation around a worksheet, navigation to a referenced worksheet, entering and editing numeric data, and entering and editing textual data. These tasks formed the basis for answering RQ2. The time to complete each of these tasks was measured and compared for both groups. It was found that the voice group took longer times for all of these tasks.

As navigation is a fundamental component of spreadsheet debugging improving navigation has the potential to improve the overall performance of users debugging spreadsheets. To this end RQ3 was constructed to identify possible ways in which navigation through voice could be improved. To answer this question the navigation of each participant, in both groups, was analysed to identify common behaviours that required multiple voice commands. Three such behaviours were observed:

- When users entered a cell with a reference to another worksheet they tended to navigate to this worksheet
- Users would audit semantically similar cells together.
- Users would skip blank cells

## 6 iVoice

Using these behaviours as a basis, an intelligent navigation system was built. This system, titled iVoice, provides a simple voice command which will perform complex navigation activities, such as moving from one cell to another cell on the same worksheet or moving between worksheets, automatically. Three features were developed:

- Move directly to a referenced Cell
- Automatically scan through a region
- Move to the next non blank cell

When looking at the navigational behaviour of all participants, it was noticed that upon initial entry of a cell with a reference to a remote cell users tended to go to that remote cell to ensure that the reference was correct. To do this using traditional voice recognition software requires users to navigate to the target worksheet through all intermediary worksheets. *iVoice*, on the other hand, allows users to skip the intermediate worksheets bringing them directly to the referenced worksheet and cell.

By analysing the formula in the current cell and assigning each referenced cell a unique colour, users can, through the command “*Jump <colour>*” where *<colour>* is the colour of the desired destination cell, move directly to the required cell. A “*Jump Back*” command is also provided to enable users to move back to the original cell. It is hypothesised that the time to reach a referenced sheet will, based upon the number of commands, decrease using the iVoice technology for non adjacent worksheets.

The list of the colours and the names used by the system can be displayed and hidden at any time through the “*Show Colours*” and “*Hide colours*” commands respectively.

The second component of the iVoice system allows users to navigate through a list of semantically similar cells quickly and easily. Semantically similar cells are those whose contents are of a similar structure and purpose. Spreadsheets are in general composed of regions of such semantically similar cells. The initial investigation has revealed that users will examine these regions sequentially, spending on average between 0.33 seconds and 1.5 seconds on each cell.

*iVoice* purposes to support this navigation activity through the provision of a scan command which automatically moves to the next cell in the chosen direction after one second. This delay allows users the chance to review the contents of the cell before moving on. To initiate this command users say “*Scan <direction>*” where *<direction>* is the direction they wish to scan, be it “left”, “right”, “up” or “down”. It is hoped that this feature will reduce the time to perform this task as it requires a single voice command rather than a series of such commands.

The third component of the iVoice system allows users to skip over blank cells. By saying “*Jump <direction>*” where *<direction>* is the way the user wants to move. The system moves directly to the next non blank cell in that direction. It is felt that this command is more efficient and natural than dictating the associated keyboard shortcut as is currently required in DrNS.

The iVoice system uses NatLink (<http://sourceforge.net/projects/natlink/>) to interface with DrNS. This application allows users to define voice commands and the associated action take upon recognition of these commands. When one of the iVoice commands is recognised the system calls a VBA (Visual Basic for Applications) macro to perform the desired action.

### 6.1 Evaluation of iVoice

When the development of iVoice was concluded, 6 experienced spreadsheet users were asked to take part in a study to evaluate the system compared to an existing voice recognition technology. The participants were required to locate, but not correct, as many errors as they could in two spreadsheets. For this they were divided into two groups, Group A and Group B. Group A were asked to audit the first spreadsheet using DrNS alone and Group B were asked to audit the same spreadsheet using DrNS and iVoice. The technologies were switched for the second spreadsheet.

The time of each cell entry and cell change was recorded during the experiment. In addition to this the time that any of the iVoice commands were used was also recorded along with the command that was used. The

overall time taken for the trial was inferred from this data. This data was collected through the T-CAT tool which was modified slightly to record the time of the iVoice commands.

By comparing the performance of the groups RQ4 was answered. For the purposes of this analysis performance was evaluated under 3 criteria: the percentage of errors found, the percentage of the spreadsheet reviewed and the total time taken. It was found that the group using iVoice found between 3% and 6% more errors than the group using DrNS alone. The average percentage of errors found ranged from 57.4% to 69.4%. Using nonparametric methods which yielded a P-value of 0.33, it was found that these results were not statistically significant.

For the purposes of measuring the percentage of the spreadsheet reviewed, only cells containing numerical or formulaic data were considered. It was determined that all other cells could be easily reviewed without being entered. On average, the users of *iVoice* covered at least 7.5% more cells than those using DrNS alone. Using nonparametric tests there is sufficient evidence to suggest these results are statistically significant. (P-value of 0.045)

Although no overall time saving was found, when a closer examination of the iVoice features was conducted, time savings were seen with each feature.

## 7 NLI for spreadsheets

As previously stated, spreadsheets can be used to store massive amounts of data. Ensuring this data is correct is essential however just as important is the ability to extract meaningful information from it. A number of tools such as AutoFilter and PivotTables can be used to extract this information however for novice spreadsheet users this can be a difficult task, especially for those using voice recognition technology. Most of these tools require multiple interactions which can be time consuming.

In similar situations, such as chart generation[Kato, Matsushita et al., 2002], Natural Language Interfaces have been used to simplify such interactions. NLI's take everyday speech, infer the desired meaning from it and perform the desired action. Kato et al[Kato, Matsushita et al., 2002], demonstrate the use of a NLI for chart generation through the following example. A user planning sales of her company's products says to the system "Show me the sales in the Shikoku district for '93 and '94". The system, in response to this, then generates a bar chart showing the sales of the Shikoku district for the years 1993 and 1994. In this example the system has eliminated the need for the user to: select the data range, select the correct chart type and enter the criteria manually.

Reducing the number of commands is not the only benefit that is offered by NLI's. NLI's have also been used to eliminate the need for complex structured commands such as those used for accessing databases. A number of studies[Zelle and Mooney R. J., 1996; Tang and Mooney, 2001] have looked at the use of NLI's to databases. Liddy et al[Liddy and Liddy, 2001] have developed such a system to assist numerous US governmental agencies in handling email requests for statistical information from the general public. Their system analyses the requests that are received and automatically generates the appropriate database query. The authors have found that their system can correctly interpret 95% of "a small sample of new user queries".

This work aims at applying a NLI to some common information extraction techniques for well structured spreadsheets. Figure 1 shows an example of a well structured spreadsheet. The spreadsheet contains a list of transactions that were completed in a number of stores. Each row of the spreadsheet contains data related to an individual sale. This spreadsheet is said to well structured as each row corresponds to one entry and each column is one type of data, as well as this, the first row of the spreadsheet contains the headings for each column.

	A	B	C	D	E	F	G
1	Date	Store	Sales Assistant	Subtotal	Vat	Sales Total	Payment Type
2	02/06/2008	Drogheda	Danny	€493.36	€103.61	€596.97	Cash
3	02/06/2008	Galway	Lisa	€157.01	€32.97	€189.98	Cheque
4	02/06/2008	Drogheda	Danny	€633.87	€133.11	€766.98	Cash
5	02/06/2008	Dublin	Nicole	€514.83	€108.12	€622.95	Cheque
6	02/06/2008	Dublin	Tommy	€213.19	€44.77	€257.96	Credit Card
7	02/06/2008	Dublin	Nicole	€138.81	€29.15	€167.96	Laser
8	02/06/2008	Dublin	Nicole	€785.93	€165.04	€950.97	Laser
9	03/06/2008	Cork	Toby	€625.59	€131.37	€756.96	Credit Card
10	03/06/2008	Dublin	Nicole	€252.04	€52.93	€304.97	Credit Card
11	03/06/2008	Dublin	Tommy	€735.50	€154.46	€889.96	Cheque
12	03/06/2008	Dublin	Nicole	€606.61	€127.39	€734.00	Laser
13	03/06/2008	Dublin	Tommy	€472.72	€99.27	€571.99	Credit Card
14	03/06/2008	Galway	James	€479.31	€100.66	€579.97	Cash
15	03/06/2008	Galway	James	€464.45	€97.53	€561.98	Laser
16	03/06/2008	Cork	Jenny	€476.83	€100.14	€576.97	Cheque
17	04/06/2008	Dublin	Tommy	€57.84	€12.15	€69.99	Credit Card
18	04/06/2008	Drogheda	Stephanie	€122.29	€25.68	€147.97	Laser
19	04/06/2008	Dublin	Tommy	€83.43	€17.52	€100.95	Cheque
20	04/06/2008	Galway	Lisa	€230.57	€48.42	€278.99	Cheque
21	04/06/2008	Cork	Toby	€21.45	€4.51	€25.96	Credit Card
22	04/06/2008	Galway	James	€399.16	€81.92	€481.08	Laser
23	04/06/2008	Drogheda	Danny	€322.31	€67.68	€389.99	Credit Card
24	04/06/2008	Galway	Lisa	€557.01	€116.97	€673.98	Cash
25	04/06/2008	Drogheda	Danny	€748.76	€157.24	€906.00	Cash
26	02/06/2008	Galway	James	€31.36	€6.59	€37.95	Cash
27	02/06/2008	Galway	James	€71.06	€14.92	€85.98	Cheque

**Figure 1: A well Structured Spreadsheet**

Details of each transaction may not be sufficient when making decisions. Most decision makers will want to know things like the total made in each store or the value of sales that were made by credit card. In order to obtain this information functions such as PivotTables and AutoFilters can be used. For novice spreadsheet users this can be a difficult task, especially for those using voice recognition technology.

To assist these users, the author proposes to use a NLI. The proposed system would allow a user to give a command such as “Show me all the credit card sales in Dublin” which would filter out all of those sales that were not made by credit card and did not take place in Dublin. This approach would eliminate the need for the user to manually go through the Store and Payment type columns applying the appropriate filters.

The first step in developing such a system is to analyse the type of utterances that would be given to such a system. For this purpose RQ5 was constructed. In order to collect these utterances, 42 students were given a brief description of the system and a sample spreadsheet and asked to write down what 10 questions they would expect such a system to be able to answer. This approach is similar to that used by Zelle[Zelle, 1995] while developing a NLI to a geographic database, geobase. Some of the utterances collected had to be removed as it was not possible to answer them based on the information in the given spreadsheet. In total 80 unique utterances were collected. These utterances were subsequently divided into groups based on the functions that would be used to answer them. Six functions were identified as being necessary to answer all questions.

Function	Number of Questions
CountIf	38
SumIf	15
Pivot Tables	11
AutoFilter	10
Max	1
Sum	2

**Table 1: Breakdown of utterances**

Table 1 shows the number of utterances that would be answered by each function. Most questions could be answered through the use of the CountIf function. These questions asked for either the number of sales with a given characteristic, for example “How many people spent more than €100?” or they would ask for the most frequent occurrence of a given characteristic such as “What was the most common form of payment?” This type of question requires an additional step which is a comparison of the count of each unique value.

The same step was also required with a number of the questions that could be answered by the SumIf function. These questions typically asked for the total when given a condition, for example “Which sales assistant had the highest sales?”

The above functions would be used when a single value was required by the user. Some of the questions however required multiple answers. Typical questions in this category include “What was the average sales for all days?” In order to present such information, a PivotTable could be used. PivotTables are used to display a summarisation of data. For example in the spreadsheet depicted in Figure 1, a pivot table could be used to show the total amount of sales for each employee for each day.

Another common request was to limit the data that was presented on screen. In most spreadsheet applications Filters can be used for this task. 10 of the questions obtained asked for only rows which met a certain criteria. One such example is "Show me the sales made by Tommy?"

The remaining functions are the sum, which would be used to answer the following type of question "What was the total amount made" and the max function which would be used when the user asked "What was the highest amount of VAT paid?"

The next step is to analyse the groups of questions. It is envisaged that common structures may be seen within each question group which can be used to determine not only the correct function to use but also the correct parameters for that function. In utterance 1 the user wants to see only those sales that were made in Dublin that had a total sales value of more than €1000 or less than €100.

- (1) Show me all of the sales in Dublin where the total sales is more than 1000 or total sales is less than 100

To achieve the desired result a filter would be applied to the store column so that only those in Dublin are shown. Then a filter would be applied to the sales total column so that only those with a value of more than 1000 or less than 100 are shown. The approach used in inferring meaning from the utterance has similarities to Woodley, Tannier et al [Woodley, Tannier et al., 2006].

The research is currently focused on recognising and implementing the Natural Language Interface. Once this is completed, RQ6 will be used to evaluate the system, through the use of a comparative study. This study will ask participants to retrieve certain information from a spreadsheet containing a large amount of data. The participants will, be split into 2 groups, Group A and Group B. Group A will firstly be asked to retrieve the information using traditional means. Group B will be asked to complete the same task using the natural language interface.

Once the participants have completed the task they will be asked to complete a questionnaire. This questionnaire will examine the participants' impressions of the NLI and also what difficulties they experienced during the experiment.

## 8 Conclusions and Acknowledgements

This paper outlines the work that has been conducted into the use of voice recognition technology for spreadsheet debugging and information retrieval. An investigative study into the use of voice recognition technology for spreadsheet debugging has shown that it is more time consuming and less efficient than conventional means. This study has highlighted navigation as being inefficient when done using voice recognition technology.

To assist in this area, the authors have developed an intelligent navigation system called *iVoice*. *iVoice* provides simple voice commands that allow users to perform complex navigational tasks, such as moving to a referenced cell on another worksheet. When using traditional voice recognition software these tasks often require multiple commands, which can be time consuming. When compared with existing voice recognition technology, users of *iVoice* found a higher percentage of the seeded errors.

This work next aims to look at the task of information retrieval. When presented with a large volume of data extracting meaningful information can be a difficult task especially for novice users who have little or no experience with the information retrieval tools commonly found in spreadsheet applications. To assist these users the authors propose the use of a Natural Language Interface, which would allow the user to tell the computer what they want in a way that is natural to them, eliminating the need to interact with the complex interfaces that are found within such tools.

This work is supported by the Irish Research Council for Science, Engineering and Technology through the embark initiative.

## References

- Bishop, B. and K. McDaid (2007). An Empirical Study of End User Behaviour in Spreadsheet Debugging. The 3rd Annual Work-in-Progress Meeting of the Psychology of Programming Interest Group.
- Bishop, B. and K. McDaid (2008). Unobtrusive Data Acquisition for Spreadsheet Research. the IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 08).
- Croll, G. J. (2005). The Importance and Criticality of Spreadsheets in the City of London. The European Spreadsheet Risk Interest Group Annual Conference, 2005.
- DeHaemer, M. J., G. Wright, et al. (1994). "Keep talking Performance effectiveness with continuous voice recognition for spreadsheet users." International Journal of Speech Technology 1(1): 41-48.
- Flood, D. and K. McDaid (2007). Voice controlled debugging of spreadsheets. The European Spreadsheet Risk Interest Group Annual Conference 2007.
- Kato, T., M. Matsushita, et al. (2002). Answering it with charts - Dialogue in natural language and charts. The 19th international conference on Computational linguistics.

- Liddy, E. and J. Liddy (2001). An NLP approach for improving access to statistical information for the masses. the FCSM 2001 Research Conference.
- Miller, G. (2006). HUD alleges overpayment for section 8. Columbia Daily Tribune.
- Panko, R. (1998). "What we know about Spreadsheet Errors." Journal of End User Computing(10): 15 - 21.
- Powell, S. G., K. R. Baker, et al. (2007). "Errors in operational spreadsheets."
- Tang, L. R. and R. J. Mooney (2001). Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing. EMCL '01: Proceedings of the 12th European Conference on Machine Learning, Springer-Verlag.
- Woodley, A., X. Tannier, et al. (2006). Natural Language Processing and XML Retrieval. The Australasian Language Technology Workshop (ALTW).
- Zelle, J. M. (1995). Using Inductive Logic Programming to Automate the Construction of Natural Language Parsers. Department of Computer Sciences. Austin, University of Texas. **Ph. D.**
- Zelle, J. M. and Mooney R. J. (1996). Learning to Parse Database Queries Using Inductive Logic Programming. The Thirteenth National Conference on Artificial Intelligence AAAI.