Final Project Write-up

John S. Chapman
4-18-2015

Executive Summary

The central purpose of this application is prototype an interface to build a dynamic model of knowledge of any topic. Traditional architectures of intelligent tutoring systems employ a model of student knowledge and a model of expert knowledge. The difference between these models is what the student has yet to learn. Since the 1970s, these models have become more and more complex resulting in a wide range of research related to how to improve the student model and the expert model, and how to compare them for differences. Artificial Intelligence has a strong hold in this arena because of the similarities between model-generation and the definition of artificial intelligence espoused by scientists for decades.

The purpose of this project within this context is to take a small step toward applying technology to improve teaching and learning. Specifically, this project takes the ideas from Conversation Theory and applies them to the design and development of an interface to build an entailment mesh. An entailment mesh is an instructional design mechanism for instructors, instructional designers, and educational technology developers. It is a way to organize knowledge that is different from current knowledge representation systems. This application helps the user participate in a knowledge generating activity either by oneself or with other users. Instructors, students, instructional designers, and educational technology developers will all find value through this tool. Instructors will find benefit in TW by building a more coherent (tight) curriculum including aligning the curriculum with course objectives and skills. Students will find value in TW by challenging what they think they know on a topic and learning from themselves and others what they did not know they did not know. Instructional designers can find value in TW by comparing the knowledge structures they have embedded in instructional products and services with the dynamic knowledge structures afford by TW. Finally, educational technology developers will discover value in TW by learning how dynamic knowledge structures will lead to improving the design of adaptive learning technologies.

Background

The field of Artificial Intelligence (AI) has produced numerous frameworks to connect knowledge together. These include Neural Nets, Semantic Nets, concept maps, and others. Marvin Minsky and others hoped to be able to structure knowledge in a way that a program can learn autonomously. Most would agree that while there have been scientific achievement related to advances in computing and innovations in technology, the efforts of artificial intelligence, compared to what the hope was, have largely been unfulfilled. I believe part of the reason for this outcome is due, in part, to the nature of knowledge. In short, there is a great debate about what knowledge is and what the resulting implications are for the design of learning and instruction.

One perspective that has been overlooked in favor of the AI wave is Conversation Theory. Conversation Theory describes knowledge as a process or action. That knowledge is dynamic. It changes regularly and it improves by being jostled or challenged. A new wave of interest in education is centered on the potential for improving education by applying technology. At the core of this endeavor is big data, analytics, and artificial intelligence. Certainly the advances made in technology even in the last 10 years suggests there are great benefits waiting for the discovery of an education-technology match that dramatically improves or extends education. In some ways, the internet has already begun this process. Information and instruction is now available at a mind-astonishing volume. Not only has quality increased, but also the availability to high quality information has increased dramatically. Wikipedia, for example, has none of the previous conceptual boundaries that encyclopedias had decades ago. However, even in an age of ubiquitous accessibility of information and instruction, education (the obtaining of a credentialed degree) still struggles to improve. Part of the problem is economics. Certainly, the higher education market is a billion dollar business tied to jobs, money, third party payers, sports, research, prestige, facilities and much, much more. There are strong influences keeping things as they are. Nevertheless, aside from the economics of higher education, what are the barriers to improving teaching and learning? Is there an opportunity to improve education-using technology? What is it? What role does technology play in improving teaching and learning?

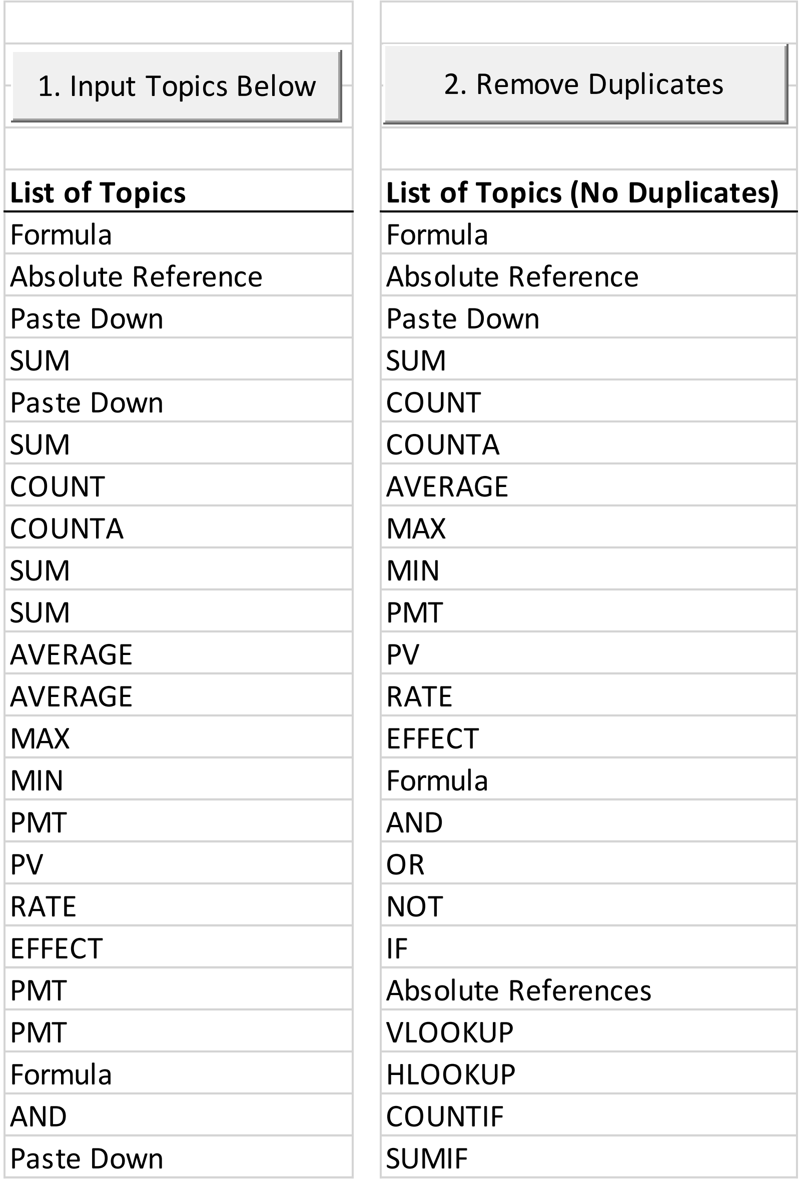
At a practical level, this project seeks to build off work done in the 70s and 80s based on Conversation Theory (CT). CT was applied to early software through a program called THOUGHTSTICKER, which is a software tool enabling the evolution of knowledge structures. THOUGHTSTICKER was built on a far, far away platform that is now defunct. This project seeks to adopt the same functions as THOUGHTSTICKER in a modern technological platform.

Implementation Documentation

The design of the application consists of three main parts. First, a concept creator engine that affords the creation of new knowledge structures. Second, a connector engine that connects concept together in different types of relationships (coherences, analogies, etc). Third, a contradiction function. This function helps identify and resolve contradictions between knowledge structures. Future development may include visualization improvement.

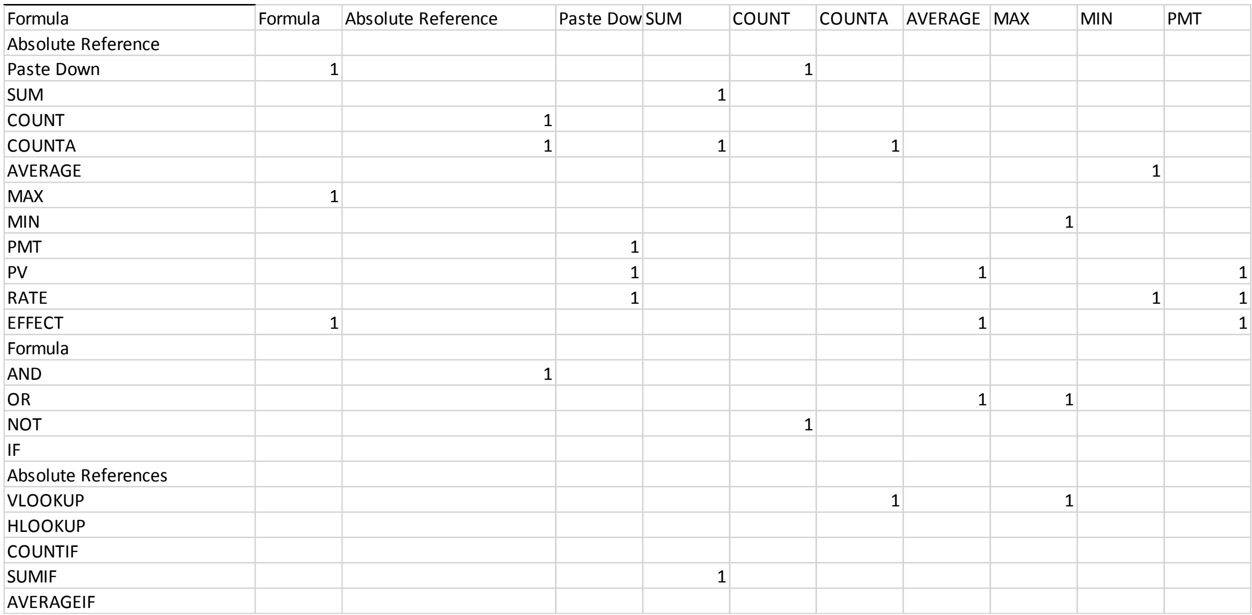
Concept Creator

The first steps in designing and building the prototype interface was to obtain the data. A previous list of knowledge concepts for an online spreadsheet course was used. But there were duplicates in the list. So, the first piece of code was to remove the duplicates. The application will work without copying a list to the file, but it also is capable of a very long list of initial topics. Below is a picture of the top portion of the initial topics, copied from a shared Google Doc. I had considered importing the data directly, but didn’t find a lot of advantage in that path mainly because the system will run with no data imported or from scratch. So, data could be imported in different ways into the application – either through entering the data manually, or copying in a list.



List of Topics	List of Topics (No Duplicates)
Formula	Formula
Absolute Reference	Absolute Reference
Paste Down	Paste Down
SUM	SUM
Paste Down	COUNT
SUM	COUNTA
COUNT	AVERAGE
COUNTA	MAX
SUM	MIN
SUM	PMT
AVERAGE	PV
AVERAGE	RATE
MAX	EFFECT
MIN	Formula
PMT	AND
PV	OR
RATE	NOT
EFFECT	IF
PMT	Absolute References
PMT	VLOOKUP
Formula	HLOOKUP
AND	COUNTIF
Paste Down	SUMIF

The second phase was designed to connect the topics together. This was the most difficult phase and was not finished in the time resources I spent for reasons I outline in the next section. However, this is the area where a lot of the synergies or benefits are experienced using Conversation Theory. The theory connects topics together so that there is a coherence between the topics. It is the achievement of a coherence that facilitates the learning. Coherence is achieved when the topics linked together are the necessary and sufficient topics to generate each other. If one topic is missing, the other topics point to, or generate the missing topic. Reaching coherence for an entire field of study or course can be intensive. But the rewards are worth the cost. The advantage of achieving a coherence of topics cannot be understated. I had planned to keep track of the relationships between the concepts by using a matrix with the same topics on both axis and a "1" to indicate a relationship with another topic. But, by the time I wrote the user form and was ready to input the data, I had not used the correct control on the form to keep track of the data. Here is what I would have liked to create:



Formula	Formula	Absolute Reference	Paste Dow	SUM	COUNT	COUNTA	AVERAGE	MAX	MIN	PMT
Absolute Reference										
Paste Down	1				1					
SUM				1						
COUNT		1								
COUNTA		1		1		1				
AVERAGE									1	
MAX	1									
MIN								1		
PMT			1							
PV			1				1			1
RATE			1						1	1
EFFECT	1						1			1
Formula										
AND		1								
OR							1	1		
NOT					1					
IF										
Absolute References										
VLOOKUP						1		1		
HLOOKUP										
COUNTIF										
SUMIF				1						
AVERAGEIF										

This picture identifies the relationships between the topics.

The third part of the design was to identify contradictions. Contradictions can take the form of topics with similar relationships. For example if two topics (SUM, and PV) were related to the same other topics (Formula and Absolute References), then there is a contradiction. Because there is no longer coherence.

Learning and Conceptual Difficulties

Getting the currently selected items to show based on the change in selection in the listbox was interesting. I found a change control that picked up any change in the listbox and then ran code to identify how many topics where selected to redim an array variable, and then created a new array with just those topics. It took me a number of iterations to fine tune the functionality because at first the text box only showed the lowest selection in the list on the left (highest index # of the listbox). So, I changed that so the textbox would capture more than one selection. Also, I needed to delete the existing text at each change. And, I had to change the settings of the textbox to allow multiple lines (or to allow carriage returns).

I also changed the modal settings of the form, so the user can interact with the excel file while the form is showing.

```
Private Sub lbTopics_Change()

Dim selTopics() As String
Dim nSelTopics As Integer
Dim i As Integer
Dim j As Integer



With lbTopics
   For i = 0 To .ListCount - 1
      If .Selected(i) Then nSelTopics = nSelTopics + 1

   Next i


ReDim selTopics(nSelTopics - 1)

j = 0

If nSelTopics > 0 Then
   For i = 0 To .ListCount - 1
      If .Selected(i) Then
         selTopics(j) = .List(i)
         Debug.Print selTopics(j)
         j = j + 1
         selectConnections (i)
      End If
   Next i

End If
```

How to keep track of the relationships between topics - it would be easy if it were a one to one or even a one to many relationship. But I need a one to two or three relationship. I think this means I need to have a relationship table where each relationship is a unique value and it can have any number of topics connected to it. and eventually, users.

Just realized I need to change a text box into a listbox on myform so I can keep track of the index value the data item has in an array. I think I can do that with a 2 column list box. But I need to change a text box into a listbox. Fairly big rewrite. But it will allow me to connect my topics in a matrx format.

After another few hours trying to troubleshoot, I realize now it would have been easier to do a 2-column listbox instead of a textbox. With the text box I would have had to format the text perfectly to search for the text in the TopicsArray() list. But because I added carriage returns to fit the array into the textbox, it is proving extremely difficult to get the carriage returns off. In short, the lessoned learned here was to not use text to search one list with another while passing the text into a text box and adding carriage returns (chr10).