Stanley Switaj
203 Artificial Intelligence
Dr. Wang

Context-Assisted Understanding

Note: This is a draft and a work in progress.

This project attempts to create an artificial intelligent system using context assisted variables.

Each variable receives the idea of the 5 W's of a plot: who, what, where, when, how or (agent giving the info), the concept, the location, the time, and how.

What makes this system unique is the flexibility to handle inconsistency and at times contradictories. As long as each concept as its own "Context Primary Key" (see below), duplicate concepts may remain in the database and remain unique.

Another interesting notion is that the robot has a limited number of engine functions to manipulate input, namely induction_f, abduction_f, and analogy_f.

Language Chosen:
The program is created in Visual Basic because of the ease in manipulating the language and the ability to use relational databases with minimal code. Prolog was considered, but I was not proficient enough in it to do anything useful since the intrinsic recursion proved elusive. Therefore a structured approach was taken with care to be as flexible as possible.

Knowledge Representation:
Every concept or keyword encountered receives an entry is a database.

Context Primary Key
The trick is that every concept receives "context variables" These context variables are a permanent signature on the concept never to be changed. The combination of the concept and the "context variables" are analogous to a unique primary key and can be summed in the following formula:

	What	Where	When	Who	How
Unique primary key =	"concept"	+ location +	time	+ speaker (input medium) +	<TBD>*

Relational Database Schema:

1) input of concepts

Ideally, one database will have a column of “concepts”. This will be matched with every word that the system comes across. Theoretically, A hash algorithm is used to place each word in the table at a particular row. For simplification, each word will be placed sequentially, but the SELECT sql statement will retrive prior stored “concepts” to check if it was placed (learned) before or not.

Each concept will have associated atomic concepts associated with it. These associated concepts are the attributes (looser) and properties (stricter) (See appendix B for distinction). Though for simplification they will be treated the same.

The theoretical database should be set up as follows:
Note = instead of “inWhat”, the term “concept” is used
Table1

ConceptID	Concept	ContextWhere	ContextWhen	ContextWho	ContextHow	fkAssConcepts	fkNotConcepts

Table2

AssConceptsID	LocationOfConcept	
	TableName+Row	

Table3

NotConceptsID	LocationOfConcept	
	TableName+Row	

The way its actually going to be implemented is that in practicality the database will combine these three tables into one to simplify coding:

Position	Variable
1	ConceptID (autonumber row number)
2	Concept (string variable)
3	ContextWhere
4	ContextWhen
5	ContextWho
6	ContextHow
7	AssConcept1
8	AssConcept1Loc_HOW_DEGREE
9	AssConcept2
10	AssConcept2Loc_HOW_DEGREE
…	AssConcept…
…	AssConcept…Loc_HOW_DEGREE
104	AssConcept49
105	AssConcept49Loc_HOW_DEGREE
106	AssConcept50
107	AssConcept50Loc_HOW_DEGREE
108	NotConcept1

109	NotConcept1Loc_HOW_DEGREE
110	NotConcept2
111	NotConcept2Loc_HOW_DEGREE
…	NotConcept…
…	NotConcept…Loc_HOW_DEGREE
205	NotConcept49
206	NotConcept49Loc_HOW_DEGREE
207	NotConcept50
208	NotConcept50Loc_HOW_DEGREE
209 – 255 variables for additional Context Associated Variables	
Example of additional context variables	
Frequency usage	
1-10 times usage (past week)	Counter (if full(value=10) R#2 increment)
R#2 Range 10 * 10 times usage (past week)	Counter (if full(value=10^2) R#3 increment)
R#3 if we get here – word is common for week (although the word may become abscure at a later contextWhen (time period)	

The idea behind each concept is that it is made up of smaller atomic values (other concepts). These concepts can be adjectives, nouns, or even ideas and each have their own distinct location.

What are the variables?

In summary, a agents knowledge is made up of the following variables:

- **Concept – the word being talked about**
- **Associated concept – a word that is associated with the concept**
- **Not associated concept – a word that limits the concept**

Example:

Concept = Desk
Associated Concept = hard
Not Assoicated Concept = rock

Note – the ideal situation is that we do not list every not associated concept, but to generate a running function to be applicable on the fly that deciphers what is and what is not. But until a certain level of intelligence to generate differences between objects, the "**NotConceptX**" will serve to provide examples of what the output of this function should be. However, an agent should keep some examples readily available to say that a concept is not a, b, c, to a reasonable n number and then be able to freely change these to other objects that are not the concept. Meaning (the agent knows a desk is not a rock, but also does a random search, finds the concept **TV** in its database, does a comparision of associated concepts (Desk to TV) and then can say that a desk is not a TV for reasons x, y, z (because the Desk Associated Concepts have few similarities to TV Associated Concepts) and/or Desk might have a "**NotConcept"** that is TV (saying that Desk is not TV explicitly).

AssConceptX
AssConcept30 = means the 30th associated concept to position variable 2 **Concept.**

AssConcept1Loc_HOW_DEGREE and
NotConcept50Loc_HOW_DEGREE

AssConcept1Loc_HOW_DEGREE is a descriptor of the concept variable right before it in the sequence of the database.
Ex. "AssConcept30" and "AssConcept30Loc_HOW_DEGREE" are related

AssConcept30Loc_HOW_DEGREE = is a parsable field containing 3 pieces of information (though they should probably be in separate fields, for simplicity sake, we combine these into one parsable varable) meaning the following:
AssConcept30Loc_HOW_DEGREE = means:

1. location that the associated context variable is located at (database name and row number), this is a learning mechanism to speed further searches
2. HOW – this is an interesting field, it attempts to summarize how the associated context is related to the field, possible summary values are
    a. IS (ISTO – is a type of/ is a category of (are) Example – abductive are relation)
    b. HAVE (a property of Example - (iHAVE) – inductive HAVE relation)
    c. ANALAGOUS TO (similar to – Example – associated concepts are similar)
    d. ETC. (other TBD)
3. DEGREE – a value of 0 to 1, and attempts to say how close the associated word is to the **Concept** (fuzzy logic, or probabilistic value)

NOTConceptX
Specificlly states that a concept is not another concept. For example, the 3 function engines might draw some conclusions, but the notConcept variables will counter.

How the system Works:

The program runs as follows:

1) A sentence is input using the ASCII character system: a text string
   For lack of a better system (one that should be based on the 5 senses of a human, but most importantly display and audio (vision and speech).

2) Each word is parsed and entered into the database.

3) The three engine functions are running to fill in words at the correct placement and attempt to integrate the knowledge

4) Where decisions are made, the system has desire or internal state variables to determine differences.

5) Ideally context variables will help create positive associations to deciper the meaning of a sentence.

6) Loop

Next logical step would be to create the same database schema, but on phrases, to get a better meaning of the sentence structure. When frequency of one word is seen with another, the **engine functions** should trigger a phrase and the associated concepts should correspond to that phrase. Also associated concepts could then be associated phrases.

Engine Functions

As stated earlier, the engine functions are induction_f, abduction_f, and analogy_f. One or all of these functions may run on a single input sentence and will alter the database to form new concepts. The new concepts may or may not be accurate, but the sum of the DEGREE of associated concepts will give a picture as to how plausible the new concept may be. Also natural language semantic will help decide which function will be used.

Example:

Induction

The induction function focuses on the concepts just input, changed, or affected, and attempts to create generalizations among other concepts.

Abduction

Will attempt to link concepts based on reasons of associated concepts as prove that concepts are related to one another.

Analogy Example:

A thingamagig is as fast as a cheetah.

Result:
Thingamagig – concept created
Given – associated concept of speed of cheetah.

Current Project Status:
Due to being hospitalized and missing time from class, the project is still being implemented and it would be merciful if it could be one of the later demonstrations.

Appendix A
Visual Basic / Access Database Limits

Microsoft Access Database Solutions - Microsoft Access Specifications and Limitations

Microsoft Access provides users with one of the simplest and most flexible RDBMS solutions on the market today. Regular users of Microsoft products will enjoy the familiar Windows 'look and feel' as well as the tight integration with other Microsoft Office family products.

As with any application, there are always specifications and limitations as to what can be done. The following details the specifications stated in the Microsoft Access Help files for Microsoft Access 2000, XP and 2003 versions.

Microsoft Access Database - General

Attribute	Maximum
Microsoft Access database (.mdb) (Microsoft Access database: A collection of data and objects (such as tables, queries, or forms) that is related to a particular topic or purpose. The Microsoft Jet database engine manages the data.) file size	2 gigabytes minus the space needed for system objects (system object: Database objects that are defined by the system, such as the table MSysIndexes, or by the user. You can create a system object by naming the object with USys as the first four characters in the object name.).
Number of objects in a database	32,768
Modules (including forms and reports with the HasModule property set to True)	1000
Number of characters in an object name	64
Number of characters in a password	14
Number of characters in a user name or group name	20
Number of concurrent users	255
Number of columns in a single database	255

Microsoft Access Table

Attribute	Maximum
Number of characters in a table name	64
Number of characters in a field name	64
Number of fields in a table	255
Number of open tables	2048; the actual number may be less because of tables opened internally by Microsoft Access
Table size	

* 2 gigabyte minus the space needed for the system objects - Acc 2003
* 2 gigabytes - Acc XP
* 1 gigabytes - Acc 2000

Number of characters in a Memo field	65,535 when entering data through the user interface; 1 gigabyte of character storage when entering data programmatically

Size of an OLE Object field 1 gigabyte
Number of indexes in a table 32
Number of fields in an index 10
Number of characters in a validation message 255
Number of characters in a validation rule 2048
Number of characters in a table or field description 255
Number of characters in a record (excluding Memo and OLE Object fields) when the UnicodeCompression property of the fields is set to Yes

* 4000 - Acc 2003
* 2000 - Acc XP

Number of characters in a field property setting 255

Microsoft Access Query
Attribute Maximum
Number of enforced relationships 32 per table minus the number of indexes that are on the table for fields or combinations of fields that are not involved in relationships
Number of tables in a query 32
Number of fields in a recordset 255
Recordset size 1 gigabyte
Sort limit 255 characters in one or more fields
Number of levels of nested queries 50
Number of characters in a cell in the query design grid 1024
Number of characters for a parameter in a parameter query 255
Number of ANDs in a WHERE or HAVING clause 99
Number of characters in an SQL statement Approximately 64,000

Microsoft Access Forms and Reports
Attribute Maximum
Number of characters in a label 2048
Number of characters in a text box 65,535
Form or report width 22 in. (55.87 cm)
Section height 22 in. (55.87 cm)
Height of all sections plus section headers (in Design View) 200 in. (508 cm)
Number of levels of nested forms or reports

* 7 - Acc 2003
* 7 - Acc XP
* 3 - Acc 2000

Number of fields or expressions you can sort or group on in a report 10
Number of headers and footers in a report 1 report header/footer; 1 page header/footer; 10 group headers/footers
Number of printed pages in a report 65,536
Number of controls and sections you can add over the lifetime of the form or report 754

Number of characters in an SQL statement that serves as the Recordsource or Rowsource property of a form, report, or control (both .mdb and .adp) 32,750

Microsoft Access Macros
Attribute Maximum
Number of actions in a macro (macro: An action or set of actions that you can use to automate tasks.) 999
Number of characters in a condition (condition: Part of the criteria that a field must meet for searching or filtering. Some conditions must be used with a value; for example, the field Author with the condition equals with the value Jane.) 255
Number of characters in a comment 255
Number of characters in an action argument (action argument: Additional information required by some macro actions. For example, the object affected by the action or special conditions under which the action is carried out.)

Appendix B
Difference between Property and Attribute

Properties and Attributes

William van den Heuvel

This page on Properties and Attributes is meant to assist in the development of the virtualistic mode of perception. The distinction between properties and attributes is useful as a mental tool to determine the virtual or real aspects of things.

Mental qualities are attributed and should therefore be called attributes. This is in contrast with physical qualities, which are intrinsic physical properties of the thing itself. So, there is an important difference between a property and an attribute. To bring out this difference more clearly, it may be useful to recall the original meaning of these words.

Property: The word property comes from Latin properius, which means "own". A property is what the thing has on its own (pertaining to itself). That means, a property denotes a objective physical aspect, which is independent of our knowledge or opinion .

Attribute: The word attribute comes from the latin verb "tribuere", which means to give. For instance, a tribute is a gift, to retribute is to give back, to attribute is to give at. An attribute is literally "given at". So, by definition, an attribute is a given fact, which has been attributed, ascribed, imputed, assigned or otherwise given. That means, by definition, an attribute is subjective; it's a virtual quality .

Attributes can only exist when there are people capable of thinking them into being. The creation and continued existence of attributes depends entirely on how and what we think. So, attributes aren't parts of the thing itself but exist only in the mind of the people who knows them.

In the common language, the words "property" and "attribute" are normally used interchangeably: This reflects the general confusion about one of the most fundamental concepts there are, which is the nature of reality. However, the very fact that the word attribute has already been in existence for a very long time, seems to indicate that early people must have been aware of attribution. This is what makes me think that the awareness about attribution and attributes has been lost at some time. Therefore, I propose to re-instate the original meaning of the words attribute and property; they can then be used again as pointers to determine whether something is real or virtual.

The words property and attribute are meant to be pointers that direct the attention to the difference. The word "property" pointing to the real aspects and "attribute" pointing to the virtual aspects of things. These pointers are very useful because they enable us to decide what is real or virtual. They are a great help in switching mentally between a realistic and a virtualistic mode of perception. Most things have both properties and attributes, which means there are always real and virtual aspects about almost anything. This applies not only to people but also to animals and objects. E.g. A person may have certain physical properties, such as blue or brown eyes, or a white or black skin, but the nationality or citizenship of that person is always a virtual attribute.

As I said, the properties are intrinsic physical qualities of the object itself: They can, in principle, be perceived through the senses or measured objectively by instruments. For instance, the height, the shape, the substance or the weight of an object are its properties. The attributes, on the other hand, are "virtues" supplied by the mind. For instance, a house may have many properties, which make it real but the value of it is an attribute. So, the value of a house is virtual. Another example is the sex of a person: Being a man or a woman is a property but being someone's husband or a wife is an attribute.

Attributes can not be perceived through the senses because they are already in the mind (after having been invented). Properties, on the other hand, do not require a mind to think them into being but they still need an observer to perceive them. One could perhaps say that properties are perceived and attributes are recognized. An attribute is a subjective interpretation or an evaluation of some kind (like the marritial status). A physicist would never think of attributes as anything real. I don't call them real either; I call them virtual.

The human mind gives (assigns) the attributes. If we didn't assign attributes, they wouldn't be there. But there will still be properties because these have not been attributed by us. The point is; by considering the difference between properties and attributes, you can determine if something is real or virtual. Properties are real (physical), whereas attributes are virtual (mental). So, if you want to change an attribute you must make a corresponding change in consciousness because that is where the attributes are.
