Journal of Northwest Semitic Languages 33/1 (2007), pp. 55-76

Jan H Kroeze (University of Pretoria)

A COMPUTER-ASSISTED EXPLORATION OF THE SEMANTIC ROLE FRAMEWORKS IN GENESIS 1:1-2:3[1]

ABSTRACT

This article focuses on the extraction and exploration of semantic role frameworks via a Visual Basic 6 program from an XML data cube containing linguistic data of the Hebrew text of Genesis 1:1-2:3. The semantic roles in the clauses were analysed and marked up according to S.C. Dik's Theory of Functional Grammar (Dik 1997a, 1997b; Kroeze 1996, 2003). The program extracts the semantic role frameworks by slicing the relevant information from a three-dimensional data cube. Unique semantic role frameworks are then identified and their frequencies calculated. The various patterns are shown and discussed. Some interesting combinations are highlighted and suggestions are made for the revision of the definitions of some of the semantic functions. The results of the study indicate the value of computer-assisted exploration for Hebrew studies and other linguistic research.

1. INTRODUCTION

One of the most frustrating tasks of a linguist is to find ample proof for a hypothesis. For example, it may consume hours, days or even weeks of working through a corpus of texts to find enough (or even one!) examples with which to illustrate an interesting construction, an unusual combination of semantic roles, or to determine the valency of a verb. Linguists' lives could be a lot easier and they could work much more efficiently if they were able to make use of proper electronic databases containing raw data on the study material. Witt (2005:79), for example, indicates the importance of using the power of computers to test hypotheses. Electronic data could also be exploited to find unknown patterns in order to suggest new hypotheses with a view to further research. "The computer can deal with far more information than you can, and even though it can't (yet) reason, it can show you opportunities for reasoning you would never find without it" (Unsworth *s.a.*).

[1] This article is a revised version of a paper read at the SASNES 2006 Conference, Unisa, 11-12 September 2006, titled "Semantic role frameworks extracted from a multidimensional XML database of Gen 1."

Linguistic databases and mark-up schemes may be regarded as examples of knowledge representation within humanities computing that may enhance productivity and discovery in humanities research (Unsworth *s.a.*). In addition, a database may capture data from various sources and therefore "the logical statements that would flow from that ontology would necessarily exceed the knowledge of any one individual" (Ramsay *s.a.*). One should, however, admit that such an ontology as reflected by a mark-up system is neither neutral nor objective and therefore limits the knowledge construction possibilities to the theoretical paradigm on which the system is based.[2] While mark-up supplied by an original author is constitutive of meaning, mark-up done to pre-existing texts is interpretive because it "... reflects the understanding of the text held by the transcriber; ... the mark-up *expresses a claim* about the text .... other interpreters might disagree" (Sperberg-McQueen *et al.* 2000:216).

In Hebrew studies, Petersen (2004), for example, developed a text database engine, called Emdros, to store and retrieve analyses of any linguistic module, facilitating advanced queries to be executed regarding texts, such as searching for examples of subject inversion and agentless passives or validating rules of agreement between subject and verb in Biblical Hebrew (BH). These types of queries are usually not possible with legacy systems. Such a system "presupposes a database which is tagged with the data necessary for answering the query." As is the case with other data-mining tools there should be interaction between the human researcher and the electronic tool since the hits found by such a search should often still be inspected to decide if they are relevant to the tested hypothesis. After all, the use of databases in humanistic sciences rather has to be regarded as a para-interpretive mechanism than a pre-interpretive one (Ramsay *s.a.*). A multi-dimensional data bank and processing system may enable the researcher to refine his/her queries (the more dimensions the better), thus reducing the amount of human analysis needed. More dimensions enable one to factor in more precise limitations for a specific search.

This study illustrates that an electronic three-dimensional data structure could be used to represent inherently multi-dimensional linguistic data regarding Biblical Hebrew clauses. The Hebrew text of Genesis 1:1-2:3 was used for the experiment.[3] To create the data for the database, the text

[2] Compare Du Plooy (1998:54, 59) for the concept of *knowledge construction*.

[3] The small data bank used for this study should be regarded as an example of data that can be used for more advanced data warehousing and data-mining

had to be analysed first. Clause by clause the text of Genesis 1:1-2:3 was transcribed,[4] translated and analysed on the different linguistic layers of morpho-syntax, syntax and semantics in an interlinear structure.[5] This data was represented in a table format where each table represents a clause and each column a phrase. The various levels of analysis were described in the various rows of the tables. Although this (typically paper-based) data bank makes perfect sense for someone who studies the work in a linear fashion, it does not facilitate synoptic (non-linear) research into linguistic structures and other phenomena. Therefore the data was transferred into an electronic data bank to make it more accessible and flexible. The most important challenge was to merge the data that resided in various, independent and unconnected tables (one for each clause) into one coherent data structure. The concept of the data structure used is a data cube, a three-dimensional collection of rows, columns and levels, integrating all the data into one data structure. The rows of the cube represent the various clauses, the columns represent the phrases of each clause, and the depth levels represent the various linguistic modules. A three-dimensional array was used in Visual Basic 6 (VB6) to capture the data, which were then programmatically tagged and saved as an XML document. The XML document can again be imported in a VB6 program for viewing, editing, extension and other processing. This process of importing and exporting an XML data bank is called round-tripping.

The creation of the database is equally important to its eventual use because the decision on what to include and exclude and how to visualize the information patterns implies underlying hermeneutics and methodologies that

projects. Ideally, *existing* Biblical Hebrew linguistic databases could, in future, be reformatted and merged into data marts constituting an integrated data warehouse, for the purpose of data investigation and knowledge creation. The XML data bank and VB program discussed in this article are available from the author.

[4] A (rough) phonetic transcription was used to make the data and results more accessible to scholars who cannot read the Hebrew script, because the same proposals could be applied to texts in any other language.

[5] These linguistic modules were chosen as examples to illustrate the integration of various linguistic perspectives into one three-dimensional data structure. If the concepts discussed in this study were applied to a complete text, such as the Hebrew Bible or a substantial part of it, other modules such as morphology and pragmatics could be added.

should be brought to light: "As with so many similar activities in digital humanities, the act of creation [of a database – JHK] is often as vital to the experiential meaning of the scholarly endeavour as the use of the final product" (Ramsay *s.a.*).

A hypertext system, such as the one proposed in this study, should be regarded as an example of humanistic informatics, which facilitates "the advancement of the understanding of human patterns of expression", thus contributing to the goal of the humanities (Aarseth *s.a.*). However, the relationship between mark-up systems and the humanities is reciprocal. Not only can the humanities benefit from them, but it can and should also contribute to their creation: "Textual scholars should not relate to mark-up technology as passive recipients of products from the computing industry, but rather be actively involved in the development and in setting the agenda, as they possess insight which is essential to a successful shaping of digital text technology" (Huitfeldt 2004).

The XML data cube described above facilitates the viewing and manipulation of the data to fulfil the needs of linguists and exegetes because the cube can be rotated or sliced and diced to reveal almost any combination of the captured data. These actions are similar to data access and manipulation facilitated by data warehousing and online analytical processing. Due to the fixed structure of the XML file they may also be regarded as a form of data retrieval "where file information is precoded for specific properties and where the conceptual categories for queries must be known in advance" (Lewis & Jones 1996). The rest of the article will discuss the extraction and evaluation of semantic role frameworks from Genesis 1:1-2:3 as a data-mining venture.

2. *EXTRACTION AND ANALYSIS OF SEMANTIC ROLE FRAMEWORKS AS A VENTURE IN TEXT DATA MINING*

Once a database of linguistic data has been created, the user is empowered to do searches for specific instances to test a theoretical possibility. XML facilitates complex searches where two or more conditions are to be true (DeRose *et al.* 1990:17). Without a proper database these are done partly manually: the researcher finds all texts that satisfy one condition and then searches within that data for the other conditions. One should remember that even if complex searches can be done (on more than one parameter) the researcher often still has to evaluate the hits returned by the database engine to see if they are relevant to verify or falsify a hypothesis (Petersen 2004).

However, the ultimate goal of a database should not only be "efficient retrieval, but interpretive insight" (cf. Ramsay *s.a.*). Data mining or "knowledge discovery" may fulfil this role. It is the process during which new patterns and trends are identified in large amounts of data. It should actually be called knowledge invention since it is a determinate process of knowledge construction. Data mining is usually done on numerical business data, but text mining is a parallel field studying the same type of activity within large amounts of text-based business data.

The type of knowledge invention illustrated in this study may be regarded as an application of text mining to humanistic resources, in this case the Hebrew text of Genesis 1:1-2:3. Although the data set is very small, it should be regarded as an experiment to explore and evaluate the applicability of these research activities for humanities computing.

Data mining, as well as text mining, forms the highest level of the knowledge invention pyramid (see Figure 1). The three lower levels work with explicit knowledge while the data-mining level pertains to implicit knowledge. In this project, the lowest level ("storing and exchanging of 'factual' knowledge") is represented by the initial capturing of the analyses of the Genesis 1:1-2:3 text using a series of two-dimensional tables.[6] The second level (modelling of 'conceptual knowledge') refers to the links between analyses of the various linguistic layers which is automatically done and implied by the data cube in the higher level of integration. If more tables were added, for example to refer to the definitions of syntactic and semantic functions,[7] the one to many relationships between these tables and the data cube could be modelled in a relational schema to show the relationships between the entity sets. The third level in the pyramid organizes and integrates information from various sources or modules in one integrated repository to facilitate online analytical processing. In this experiment this phase is represented by the building of the data cube either in XML or in the three-dimensional array in VB6. The cube may be regarded as a multi-dimensional data warehouse. Multi-dimensional interpretation is a form of data mining. Therefore, extracting patterns, such as unique semantic role frameworks, from the cube is an implementation of knowledge invention. It consciously explores the integrated data in order to find new combinations or, at least, to confirm existing hypotheses about known

[6] The analysis itself is, of course, the explication of implicit knowledge.

[7] Semantic function is a synonym for semantic role.

combinations.[8] Even if no new combinations are found, knowledge is still created, namely that there are no new patterns, which may confirm an existing theory. If no examples of a known pattern are found, it may indicate that the pattern is very rare, but one would need a comprehensive data set (e.g. of the whole Hebrew Bible) to conclude that a certain pattern does not occur at all. These insights may be regarded as "the uncovering of *new, implicit* and *potentially useful* knowledge from large amounts of data" (Cannataro *et al.* 2002:33-34).

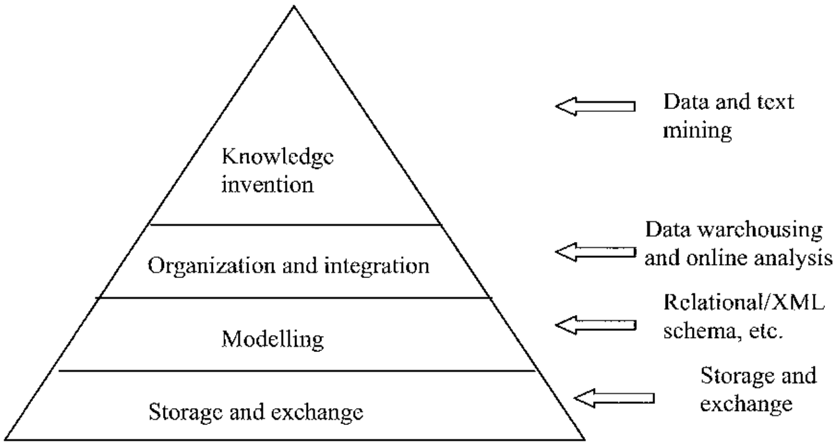
Knowledge invention
Organization and integration
Modelling
Storage and exchange
Data and text mining
Data warehousing and online analysis
Relational/XML schema, etc.
Storage and exchange

Figure 1. The knowledge invention paradigm (adapted from Cannataro *et al.* 2002:34).

Wintner (2004:128) identified a lack of Hebrew computational linguistic projects that deal directly with semantics. This article tries to make a small contribution to fill that gap. Embarking on such a venture is, however, no easy task, since human language is a complex phenomenon, especially in the case of Hebrew "due to its rich morphology and deficient script" (Wintner 2004:113-114). For this experiment these additional complexities were eliminated by using transcription for the rendering of the Hebrew text and ignoring the morphological module.

[8] For another example of using XML to integrate linguistic data, compare Bird *et al.* (2002) who converted divergent lexicographical material from three South-west USA languages to the same format.

The program logic used to extract and analyse the relevant information is discussed below, although the program code will not be rendered here. The following steps are followed:

- Slice off the semantic functions module
- Sort the elements in each row
- Concatenate each row
- Order the rows
- Identify unique frameworks
- Calculate the frequencies of unique frameworks

The program uses, *i.a.*, for-loops and if-statements to process the data in the three-dimensional array. These functionalities (repetition structures and conditions) are also typical of template-based XML query languages (Bourret 2003). While the array structure itself facilitates the data storage, the array manipulation should be regarded as an application tool that enables data mining. Storage and application tools are two of the main data warehouse modules. Application tools include online analytical processing (OLAP), mining and analysis tools (Wang & Dong 2001:49).

3. SLICING OFF THE SEMANTIC FUNCTIONS LAYER FROM THE DATA CUBE

To slice off from the data cube the dimension containing information on the semantic functions, a subset of the data is copied from the three-dimensional data structure into a two-dimensional array. Each row represents a unique clause in linear order of appearance. The columns represent the successive phrases of the clauses. The semantic functions of these phrases are recorded in the intersecting cells. While this array is being populated with the semantic functions from the data cube, validation is performed to ensure that the terminology used is consistent. Numbers are prefixed to the semantic functions to facilitate a logical sorting process of the rows' elements and the rows themselves (as opposed to an alphabetical ordering according to the names of the functions). If no semantic function is present, indicated by a dash, it is numbered as 98 to ensure that it will be placed after all the other semantic functions later when ordering occurs. If an invalid semantic function is found (e.g. incorrectly spelled), an error message is shown. The user should correct the source data before continuing with the ensuing steps. When this part of the program is run, the semantic role frameworks of the 108 clauses are shown as the resulting output (see Table 1).

```
SLICE OFF SEMANTIC FUNCTION FRAMES[9] FROM GEN 1 DATA CUBE

Clause 1:  + 23. Time + 01. Action + 05. Agent + 10a. Patient + 98. - + Gen01v01a
Clause 2:  + 09. Zero + 04. State + 33. Classification + 98. - + 98. - + Gen01v02a
Clause 3:  + 09. Zero + 12. Location + 98. - + 98. - + 98. - + Gen01v02b
Clause 4:  + 06. Positioner + 02. Position + 12. Location + 98. - + 98. - + Gen01v02c
Clause  5:  + 01. Action + 05. Agent + 10a. Patient + 98. - + 98. - + Gen01v03a
Clause  6:  + 04. State + 09. Zero + 98. - + 98. - + 98. - + Gen01v03b
Clause  7:  + 04. State + 09. Zero + 98. - + 98. - + 98. - + Gen01v03c
Clause  8:  + 03. Process + 08. Processed + 10a. Patient + 98. - + 98. - + Gen01v04a
Clause  9:  + 09. Zero + 98. - + 34. Quality + 98. - + 98. - + Gen01v04b
Clause  10:  + 01. Action + 05. Agent + 10a. Patient + 14. Source + 98. - + Gen01v04c
Etc....
Clause  108:  + 10b. Product + 01. Action + 05. Agent + 19. Manner + 98. - + Gen02v03d
```

Table 1. Semantic role frameworks extracted from the three-dimensional data bank.

Although these results are not very user-friendly it is only the first, but essential step in the data-mining process. The following steps will use this subset of data to organise and analyse the data. Aggregating the extracted combinations of semantic roles into groups of unique semantic role frameworks may be regarded as the identification of a pattern or patterns, which is one of the characteristics of data mining (Thuraisingham 2002:191). This process is not only based on a linguistic knowledge representation (the marked-up text) but also creates, or at least enhances, a new knowledge representation (the patterns of semantic role frameworks). The outcomes of humanistic computing is the explication of existing knowledge about a discipline by means of a knowledge representation, testing this ontology against a body of (marked-up) data and the creation of new insights: "[C]omputing a representation of large amounts of material that has been encoded and processed according to a rigorous, well thought-out knowledge representation affords opportunities for perceiving and analysing patterns, conjunctions, connections, and absences that a human being, unaided by a computer, would not be likely to find" (Unsworth *s.a.*).

4. SORTING THE ELEMENTS IN EACH ROW

To facilitate aggregation functions the data should be sorted logically first. First, the numbers appended to the semantic functions are used to

[9] In this article *frameworks* and *frames* are used as synonyms.

sort the semantic functions in each clause in the order of clause type, arguments and satellites. Although the word order is lost, the goal here is to deduct the logical structure of the semantic role frameworks. If word order is important for the researcher it could be recovered by means of the clause references. When this part of the program is run, the semantic functions are logically ordered per clause (see Table 2).

```
SEMANTIC FUNCTION FRAMES ORDERED LOGICALLY USING NUMBERS

Clause 1:  + 01. Action + 05. Agent + 10a. Patient + 23. Time + 98. - + Gen01v01a
Clause 2:  + 04. State + 09. Zero + 33. Classification + 98. - + 98. - + Gen01v02a
Clause 3:  + 09. Zero + 12. Location + 98. - + 98. - + 98. - + Gen01v02b
Clause 4:  + 02. Position + 06. Positioner + 12. Location + 98. - + 98. - + Gen01v02c
Clause 5:  + 01. Action + 05. Agent + 10a. Patient + 98. - + 98. - + Gen01v03a
Clause 6:  + 04. State + 09. Zero + 98. - + 98. - + 98. - + Gen01v03b
Clause 7:  + 04. State + 09. Zero + 98. - + 98. - + 98. - + Gen01v03c
Clause 8:  + 03. Process + 08. Processed + 10a. Patient + 98. - + 98. - + Gen01v04a
Clause 9:  + 09. Zero + 34. Quality + 98. - + 98. - + 98. - + Gen01v04b
Clause 10:  + 01. Action + 05. Agent + 10a. Patient + 14. Source + 98. - + Gen01v04c
Etc.
```

Table 2. Extracted semantic role frameworks ordered logically by the semantic roles of clause types, arguments and satellites.

These results are still not good enough since the researcher still has to do most of the analysis him-/herself. One now needs to order the rows as units with reference to each other.

5. *CONCATENATION OF EACH ROW*

The collections of semantic functions in each row now have to be concatenated into single strings to facilitate sorting of the frameworks (regarded as units, similar to a set of words), implying that the two dimensions (clauses and semantic functions) are merged into one. However, the clause numbers and primary keys must be stored in a parallel dimension for reference purposes. The sorting is done only on the semantic role frameworks. The dummy number 98 to mark lacking functions is removed before sorting since all the dashes have already been moved to the end of the collections. (Not all clauses have five phrases, nor do all phrases have semantic functions.)

The output appears to be very similar to that of the previous step, but the underlying structure is actually quite different because the semantic role frameworks have been merged before the printing phase into single strings, one for each clause. Each clause's framework is now similar to a single word which can be ordered easily.

6. *ORDERING THE ROWS AS UNITS WITH REFERENCE TO EACH OTHER*

The next step is to order the rows of semantic role frameworks which have been merged into single strings. Running this section of the program produces the following output showing that identical (and similar) frameworks are now ordered contiguously (see Table 3).

```
SEMANTIC FUNCTION FRAMES OF ALL CLAUSES ORDERED LOGICALLY WITH
REGARD TO EACH OTHER

01. Action + - + - + - + -. Reference: Gen01v28f (Clause 86)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v03a (Clause 5)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v06a (Clause 15)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v07a (Clause 18)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v09a (Clause 26)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v11a (Clause 34)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v14a (Clause 44)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v20a (Clause 55)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v24a (Clause 69)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v26a (Clause 75)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v28a (Clause 81)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v28b (Clause 82)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen01v29a (Clause 88)
01. Action + 05. Agent + 10a. Patient + - + -. Reference: Gen02v03a (Clause 105)
01. Action + 05. Agent + 10a. Patient + 10b. Product + -. Reference: Gen01v05a (Clause 11)
01. Action + 05. Agent + 10a. Patient + 10b. Product + -. Reference: Gen01v08a (Clause 23)
01. Action + 05. Agent + 10a. Patient + 10b. Product + -. Reference: Gen01v10a (Clause 30)
Etc.
```

Table 3. Extracted semantic role frameworks concatenated and ordered as units with reference to each other.

This output is already usable in a small dataset, but if a larger dataset were used, it would become verbose and clumsy due to the repetition of similar frameworks.

7. *IDENTIFYING AND COUNTING UNIQUE SEMANTIC ROLE FRAMEWORKS*

The last two steps in analysing the semantic role frameworks are to identify unique combinations and to calculate the frequencies of each unique framework. This statistical data may be used to indicate which patterns occur frequently or rarely. The program uses parallel arrays to store information on the unique frameworks, their frequencies and references to the individual instances of each. This process forces the researcher to look at his/her data in a very rigorous way, thus revealing inconsistencies that might still exist in the tagged text. This is indeed the case in this exemplary analysis of Genesis 1:1-2:3. In order to demonstrate this feature, these inconsistencies have not yet been corrected in the tagged data bank or in the results below. The results produced by running this section of the program are shown in the left column of Table 4. One example of each framework is provided in the right-hand column.

UNIQUE SEMANTIC FUNCTION FRAMES, NUMBER OF OCCURRENCES AND CLAUSE REFERENCES	**EXAMPLES OF FRAMES AS FOUND IN CLAUSES**
Frame 1: 01. Action + - + - + - + - Number of occurrences: 1 References: Gen01v28f (Clause 86)	Gen 1:28f: *vexif$uha* and rule it
Frame 2: 01. Action + 05. Agent + 10a. Patient + - + - Number of occurrences: 13 References: Gen01v03a (Clause 5) Gen01v06a (Clause 15) Gen01v07a (Clause 18) Gen01v09a (Clause 26) Gen01v11a (Clause 34) Gen01v14a (Clause 44) Gen01v20a (Clause 55) Gen01v24a (Clause 69) Gen01v26a (Clause 75) Gen01v28a (Clause 81) Gen01v28b (Clause 82) Gen01v29a (Clause 88) Gen02v03a (Clause 105)	Gen 1:28a: *vayvarex otam elohim* and God blessed them
Frame 3: 01. Action + 05. Agent + 10a. Patient + 10b. Product + - Number of occurrences: 3 References: Gen01v05a (Clause 11) Gen01v08a (Clause 23) Gen01v10a (Clause 30)	Gen 1:5a: *vayiqra elohim la'or yom* and God called the light day
Frame 4: 01. Action + 05. Agent + 10a. Patient + 12. Location + 28. Purpose Number of occurrences: 1 References: Gen01v17a (Clause 50)	Gen 1:17a: *vayiten otam elohim birkia ha$amayim leha'ir al ha'arec ...* and God gave them in the firmament of the heaven to shine on the earth ...

Frame 5: 01. Action + 05. Agent + 10a. Patient + 14. Source + - Number of occurrences: 1 References: Gen01v04c (Clause 10)	Gen 1:4c: *vayavdel elohim ben ha'or uven haxo$ex* and God separated between the light and between the darkness (*i.e.* the light away from the darkness)
Frame 6: 01. Action + 05. Agent + 10a. Patient + 19. Manner + - Number of occurrences: 1 References: Gen01v22a (Clause 62)	Gen 1:22a: *vayvarex otam elohim lemor ...* and God blessed them by saying ...
Frame 7: 01. Action + 05. Agent + 10a. Patient + 23. Time + - Number of occurrences: 2 References: Gen01v01a (Clause 1) Gen02v02a (Clause 101)	Gen 1:1a: *bre$it bara elohim et ha$amayim ve'et ha'arets* in the beginning God created the heaven and the earth
Frame 8: 01. Action + 05. Agent + 10b. Product + 19. Manner + - Number of occurrences: 4 References: Gen01v21a (Clause 58) Gen01v25a (Clause 72) Gen01v27a (Clause 78) Gen02v03d (Clause 108)	Gen 1:27a: *vayivra elohim et ha'adam becalmo* and God created the people in his image
Frame 9: 01. Action + 05. Agent + 10b. Product + 28. Purpose + - Number of occurrences: 1 References: Gen01v16a (Clause 49)	Gen 1:16a: *vaya'as elohim et $ney ham'orot hagdolim et hama'or hagadol ... ve'et hama'or hakaton ... ve'et hakoxavim lemem$elet hayom ... lemem$elet halayla* and God made the two great lights, the big light ..., and the small light ..., and the stars to the dominion of the day ... to the dominion of the night
Frame 10: 01. Action + 05. Agent + 12. Location + - + - Number of occurrences: 1 References: Gen01v21b (Clause 59)	Gen 1:21b: *a$er $arcu hamayim* that swarms (in) the water
Frame 11: 01. Action + 05. Agent + 12. Location + 12. Location + - Number of occurrences: 1 References: Gen01v20c (Clause 57)	Gen 1:20c: *ve'of ye'ofef al ha'arec al pney rekia ha$amayim* and let flying creatures fly over the land in front of the firmament of the heaven
Frame 12: 01. Action + 10a. Patient + - + - + - Number of occurrences: 4 References: Gen01v22d (Clause 65) Gen01v28e (Clause 85) Gen01v28g (Clause 87) Gen02v01a (Clause 100)	Gen 1:22d: *umil'u et hamayim bayamim* and fill the water in the sea

Frame 13: 01. Action + 10a. Patient + 10b. Product + - + - Number of occurrences: 2 References: Gen01v05b (Clause 12) Gen01v10b (Clause 31)	Gen 1:5b: *velaxo$ex qara layla* and to the darkness He called night
Frame 14: 01. Action + 10a. Patient + 11. Receiver + - + - Number of occurrences: 1 References: Gen01v29b (Clause 89)	Gen 1:29b: *hineh natati laxem et kol esev zorea zera ...* behold I give to you all green plants yielding seed ...
Frame 15: 01. Action + 10a. Patient + 14. Source + - + - Number of occurrences: 2 References: Gen01v06c (Clause 17) Gen01v07b (Clause 19)	Gen 1:7b: *vayavdel ben hamayim ... uven hamayim ...* and He separated between the water ... and between the water ...
Frame 16: 01. Action + 10a. Patient + 23. Time + - + - Number of occurrences: 2 References: Gen02v02c (Clause 103) Gen02v03c (Clause 107)	Gen 2:2c: *vayi$bot bayom ha$vi'i mikol melaxto ...* and He stopped on the seventh day with all his work ...
Frame 17: 01. Action + 10a. Patient + 29. Reason + - + - Number of occurrences: 1 References: Gen02v03b (Clause 106)	Gen 2:3b: *vaykade$ oto [ki vo $avat mikol melaxto ...]* and He consecrated it [because on it He stopped with all his work ...]
Frame 18: 01. Action + 10b. Product + - + - + - Number of occurrences: 3 References: Gen01v31b (Clause 96) Gen02v02b (Clause 102) Gen02v02d (Clause 104)	Gen 1:31b: *a$er asa* that He made
Frame 19: 01. Action + 10b. Product + 19. Manner + - + - Number of occurrences: 2 References: Gen01v26b (Clause 76) Gen01v27b (Clause 79)	Gen 1: 27b: *becelem elohim bara oto* in the image of God He created him
Frame 20: 01. Action + 10b. Product + 34. Quality + - + - Number of occurrences: 1 References: Gen01v27c (Clause 80)	Gen 1:27c: *zaxar unkeva bara otam* male and female He created them
Frame 21: 02. Position + 06. Positioner + 12. Location + - + - Number of occurrences: 1 References: Gen01v02c (Clause 4)	Gen 1:2c: *veruach elohim meraxefet al pney hamayim* and the spirit of God (was) hovering on the surface of the water

Frame 22: 03. Process + - + - + - + - Number of occurrences: 2 References: Gen01v22c (Clause 64) Gen01v28d (Clause 84)	Gen 1:22c: *urvu* and become numerous
Frame 23: 03. Process + 08. Processed + - + - + - Number of occurrences: 1 References: Gen01v09c (Clause 28)	Gen 1:9c: *vetera'eh hayaba$a* and let the dry land become visible
Frame 24: 03. Process + 08. Processed + 10a. Patient + - + - Number of occurrences: 7 References: Gen01v04a (Clause 8) Gen01v10c (Clause 32) Gen01v12c (Clause 40) Gen01v18b (Clause 51) Gen01v21c (Clause 60) Gen01v25b (Clause 73) Gen01v31a (Clause 95)	Gen 1:31a: *vayar elohim et kol ...* and God saw everything ...
Frame 25: 03. Process + 08. Processed + 10b. Product + 19. Manner + - Number of occurrences: 2 References: Gen01v12a (Clause 38) Gen01v24b (Clause 70)	Gen 1:12a: *vatoceh ha'arec de$e esev mazria zera ... ve'ec ose pri ... leminehu ... leminehu* And the land produced grass, plant(s) bearing fruit ... and tree(s) making fruit ... according to its kind ... according to its kind
Frame 26: 03. Process + 08. Processed + 12. Location + - + - Number of occurrences: 2 References: Gen01v09b (Clause 27) Gen01v22e (Clause 66)	Gen 1:9b: *yikavu hamayim mitaxat ha$amayim el makom exad* let the water collect under the heaven to one place
Frame 27: 03. Process + 08. Processed + 15. Reference + - + - Number of occurrences: 2 References: Gen01v11b (Clause 35) Gen01v20b (Clause 56)	Gen 1:20b: *yi$recu hamayim $erec nefe$ xaya* let the water swarm (with) swarming things, living beings
Frame 28: 04. State + - + - + - + - Number of occurrences: 2 References: Gen01v22b (Clause 63) Gen01v28c (Clause 83)	Gen 1:22b *pru* be fruitful

Frame 29: 04. State + 09. Zero + - + - + - Number of occurrences: 8 References: Gen01v03b (Clause 6) Gen01v03c (Clause 7) Gen01v05c (Clause 13) Gen01v08b (Clause 24) Gen01v13a (Clause 42) Gen01v19a (Clause 53) Gen01v23a (Clause 67) Gen01v31d (Clause 98)	Gen 1:3c: *vayehi or* and there was light
Frame 30: 04. State + 09. Zero + 12. Location + - + - Number of occurrences: 1 References: Gen01v06b (Clause 16)	Gen 1:6b: *yehi rakia betox hamayim* let there be a firmament in the heaven
Frame 31: 04. State + 09. Zero + 12. Location + 28. Purpose + - Number of occurrences: 1 References: Gen01v14b (Clause 45)	Gen 1:14b *yehi me'orot birkia ha$amayim lehavdil ben hayom uven halayla* let there be lights in the firmament of the heaven to separate between the day and between the night
Frame 32: 04. State + 09. Zero + 33. Classification + - + - Number of occurrences: 1 References: Gen01v02a (Clause 2)	Gen 1:2a: *veha'arets hayta tohu vavohu* and the earth was an emptiness and void
Frame 33: 04. State + 09. Zero + 34. Quality + - + - Number of occurrences: 6 References: Gen01v05d (Clause 14) Gen01v08c (Clause 25) Gen01v13b (Clause 43) Gen01v19b (Clause 54) Gen01v23b (Clause 68) Gen01v31e (Clause 99)	Gen 1:5d: *vayehi voker yom exad* and it was morning day one
Frame 34: 04. State + 12. Location + 28. Purpose + 33. Classification + - Number of occurrences: 1 References: Gen01v15a (Clause 47)	Gen 1:15a: *vehayu lim'orot birkia ha$amayim leha'ir al ha'arec* and let them be to lights in the firmament of the heaven to shine on the earth
Frame 35: 04. State + 16. Beneficiary + 28. Purpose + - + - Number of occurrences: 1 References: Gen01v29c (Clause 92)	Gen 1:29e *laxem yiheyeh le'oxla* to you it will be as food

Frame 36: 04. State + 19. Manner + - + - + - Number of occurrences: 6 References: Gen01v07e (Clause 22) Gen01v09d (Clause 29) Gen01v11d (Clause 37) Gen01v15b (Clause 48) Gen01v24c (Clause 71) Gen01v30c (Clause 94)	Gen 1:7e: *vayehi xen* and it was so
Frame 37: 04. State + 33. Classification + - + - + - Number of occurrences: 1 References: Gen01v14c (Clause 46)	Gen 1:14c: *vehayu le'otot ulmo'adim ulyamim ve$anim* and let them be to signs and to seasons and to days and years
Frame 38: 05. Agent + 10a. Patient + - + - + - Number of occurrences: 1 References: Gen01v26c (Clause 77)	Gen 1:26c: *veyirdu vidgat hayam uv'of ha$amayim uvabhema uvxol ha'arec uvxol haremes haromes al ha'arec* and let them govern over the fish of the sea and over the flying creatures of the heaven and over the animals and over the whole earth and over all the small animals that swarm on the earth
Frame 39: 09. Zero + 09. Zero + 12. Location + - + - Number of occurrences: 1 References: Gen01v29d (Clause 91)	Gen 1:29d: *a$er bo pri ec zorea zara* in which (there is) fruit of seed bearing trees
Frame 40: 09. Zero + 12. Location + - + - + - Number of occurrences: 7 References: Gen01v02b (Clause 3) Gen01v07c (Clause 20) Gen01v07d (Clause 21) Gen01v11c (Clause 36) Gen01v12b (Clause 39) Gen01v29c (Clause 90) Gen01v30b (Clause 93)	Gen 1:2b: *wexo$ex al pney tehom* and darkness (was) on the surface of (the) primeval ocean
Frame 41: 09. Zero + 34. Quality + - + - + - Number of occurrences: 1 References: Gen01v04b (Clause 9)	Gen 1:4b: *et ha'or ki tov* that the light (is) good

Frame 42: 34. Quality + - + - + - + - Number of occurrences: 6 References: Gen01v10d (Clause 33) Gen01v12d (Clause 41) Gen01v18d (Clause 52) Gen01v21d (Clause 61) Gen01v25c (Clause 74) Gen01v31c (Clause 97)	Gen 1:10d: *ki tov* that (it was) good

Table 4. Unique semantic role frameworks found in Gen 1:1-2:3.

These results provide the researcher with good information which may now be used to confirm of falsify existing hypotheses, or to create new ones.

The following frameworks were to be expected and confirm existing definitions of semantic functions: Frames 3, 4, 5, 6, 8, 10, 21, 23, 26, 27, 29, 30, 32, 40, 41.

The following frameworks are interesting, but do not contradict existing definitions of semantic functions:

- Two location satellites: Frame 11
- Patient after process verb: Frame 24
- Zero assumed (elliptic): Frame 42

The following frameworks are interesting, but at closer inspection reveal deficiencies in the tagging scheme and prompt better (more detailed) coding:

- Agent included in action verb: Frames 12-20
- Agent and product included in action verb: Frame 1
- Processed included in process verb: Frame 22
- Zero included in state verb: Frames 28, 34-37

While working through the data according to the identified patterns a number of analysis and tagging mistakes or inconsistencies were found. These should be corrected as an outcome of this article.[10]

- Frame 38 is incorrectly tagged: "Agent" should be replaced with "Action." Verse 1:26c is actually another example of Frame 12, which itself should be tagged in more detail.
- The semantic function of direct speech as direct object should be changed from patient to product in verses 1:3a, 1:6a, 1:7a, 1:9a, 1:11a, 1:14a, 1:20a, 1:24a, 1:26a, 1:28b, 1:29a, 2:3a (cf. Frame 2)

[10] If such a corrected version were used as source data for the analysis of semantic role frameworks the results (including the numbering of the unique frames) would change.

- The semantic function of patient should be changed to product in verse 1:1a (cf. Frame 7)
- Inconsistencies regarding the tagging format of embedded phrases should be corrected: see verses 1:2a, 1:16a, 1:21a, 1:24b, 1:25a (cf. Frames 8, 9, 25)
- Inconsistencies regarding the tagging format of embedded clauses should be corrected: see verses 1:3a, 1:4a, 1:11c, 1:17a-1:18a, 1:29b (cf. Frames 2, 4, 14, 24, 40)
- Processed should be replaced by force in verses 1:12a and 1:24b (cf. Frame 25)
- Zero should be changed to classification in verses 1:5d, 1:8c, 1:13b, 1:19b, 1:23b, 1:31e (cf. Frame 33)
- Purpose should be changed to classification in verses 1:16a, 1:29e (cf. Frame 9, 35)
- The zero allocated to a relative pronoun should be changed to a dash (-) in verse 1:29d (cf. Frame 39)

The most important outcome of the data-mining venture is that a small number of frameworks were found that contradict existing definitions of semantic functions:

- **Purpose in state:** Frames 31, 34 (vs. 1:14b, 1:15a)

According to Dik (1997a:244) a purpose satellite can only be found in a controlled predication. The examples found in Gen 1:14b and 15a ("to be/exist in order to separate/shine") clearly shows that in BH a state predicate may contain an embedded predication with the semantic function of purpose.

- **Quality after product:** Frame 20 (vs. 1:27c)

According to Dik (1997a:214) the semantic function of "property assignment" (here called "quality") is restricted to non-verbal predicates (adjectives or bare nominals). One should, however, consider assigning this role also to some adjectival phrases, such as in this example ("*male and female* he created them").

- **Quality after classification:** Frame 33 (vs. 1:5d, 1:8c, 1:13b, 1:19b, 1:23b, 1:31e)

These examples are similar to the previous case, since they contain phrases in apposition which are regarded as adjectival modifiers. Dik (1997a:62) would regard these examples as term operators or restrictors. More loosely coupled than mere attributes, they highlight the need to assign the quality role to phrases in apposition (e.g. "and it was morning, *day one*").

- **Manner in state:** Frame 36 (vs. 1:7e, 1:9d, 1:11d, 1:15b, 1:24c, 1:30c)

According to Dik (1997a:230) manner satellites occur in actions, positions and processes. Although this recurring example ("and it was so") is not a very strong one, it does prompt the grammarian to reconsider the possibility of manner occurring in states as well.

These instances should be reconsidered and either the tagging should be changed or the definitions of the relevant semantic functions should be adjusted. It may be concluded that the patterns identified by the data-mining process proved to be very useful in the knowledge invention process of checking and refining the definitions of semantic functions.

8. CONCLUSION

The process of creating the data bank, i.a. by marking up the phrases' semantic functions according to Dik's theory, not only shows the suitability of the Theory of Functional Grammar for Biblical Hebrew, but also the feasibility of creating an exploitable electronic data bank of linguistic data.

A computer-assisted exploration of the semantic data captured in the XML data bank of Gen 1:1-2:3 illustrated the significance of such a venture. By enabling the researcher to look at the data from a different perspective, inconsistencies in the tagging of Gen 1:1-2:3 were discovered. This proves the usability of using a computer to assist the human researcher by enforcing a certain level of rigour that is difficult or even impossible for humans to acquire. This rigour not only helps researchers to discover their own mistakes, but could also lead to new insights. In the words of Unsworth (*s.a.*): "The process that one goes through in order to develop, apply and compute these knowledge representations is unlike anything that humanities scholars, outside of philosophy, have ever been required to do. This method, or perhaps we should call it a heuristic, discovers a new horizon for humanities scholarship, a paradigm as powerful as New Criticism, New Historicism, or Deconstruction – indeed, very likely more powerful, because the rigor it requires will bring to our attention undocumented features of our own ideation, and coupled with enormous storage capacity and computational throughput, this method will present us with patterns and connections in the human record that we would otherwise never have found or examined."

Even in this small experiment some new patterns regarding combinations of semantic functions emerged, for example, a purpose embedded within a state predication. Although the results of the data-mining process provided overall support for the applicability of Dik's theory for a Hebrew text, it also revealed possible deficiencies in the definitions of some semantic functions, at least when these are applied to BH. It is rather surprising that, in the limited dataset used for this experiment, the discovery was made of four semantic role frameworks that contradict existing definitions of semantic functions. The discovery of such deficiencies could help grammarians to formulate and research new hypotheses, such as revising the definitions of semantic functions for specific languages. For example, the deficiencies referred to above should be researched using other languages as source data to determine if they are Hebrew specific or more general in nature. Were the dataset larger one could probably expect many similar discoveries that prompt processes of knowledge invention. This again highlights the need for extensive linguistic databases. Humanities scholars should regard the encoding of their deep knowledge of texts to build XML software as "an investment in the longer-term future of their discipline" – in order to leave a persistent and usable legacy for succeeding generations (Flynn 2002:59).

Digital data banks like this are not only useful for information system-oriented activities such as text data warehousing and text data mining, but similar studies are also essential to build large tagged linguistic corpora needed by researchers of natural language processing. They can use these data banks to train computer programs to simulate human language understanding and creation. Hebrew linguists can make a huge contribution to enhance research in natural language processing by fulfilling this need as identified by Wintner (2004:131): "Preparation of such corpora is a long, arduous and expensive process, requiring mostly the labor of computational linguists with sufficient knowledge of the language and its syntax and at least some computational background. Undoubtedly such corpora for Hebrew are extremely important and well worth investing in."

BIBLIOGRAPHY

Aarseth, E *s.a.* The Field of Humanistic Informatics and its Relations to the Humanities. *Essays in Humanities Computing.* [Online.] Available: http://www.digitalhumanities.org/Essays/ [Cited 23 November 2005].

Bird, S, Hammond, M, Amarillas, M, Jeffcoat, M, Harley, H, Miyashita, M, Moll, L, Willie, MA & Zepeda, O 2002. Web-Based Dictionaries for Languages of the South-West USA. *Literary and Linguistic Computing* 17(4), 427-438.

Bourret, R 2003. *XML and Databases.* [Online.] Available: http://www.rpbourret.com/xml/XMLAndDatabases.htm [Cited 20 October 2003].

Cannataro, M, Guzzo, A & Pugliese, A 2002. Knowledge Management and XML: Derivation of Synthetic Views over Semistructured Data. *ACM Sigapp Applied Computing Review* 10(1), 33-36.

DeRose, S J, Durand, D G, Mylonas, E & Renear, A H 1990. What Is Text, Really? *Journal of Computing in Higher Education* 1(2), 3-26.

Dik, S C 1997a. *The Theory of Functional Grammar. Part 1. The Structure of the Clause* (edited by Kees Hengeveld). 2nd ed. Berlin: Mouton de Gruyter.

Dik, S C 1997b. *The Theory of Functional Grammar. Part 2. Complex and Derived Constructions* (edited by Kees Hengeveld). Berlin: Mouton de Gruyter.

Du Plooy, N F 1998. An Analysis of the Human Environment for the Adoption and Use of Information Technology. D.Com. thesis, University of Pretoria.

Flynn, P 2002. Is there Life beyond the Web? *Literary and Linguistic Computing* 17(1), 49-59.

Huitfeldt, C 2004. Scholarly Text Processing and Future Mark-up Systems. *Essays in Humanities Computing.* [Online.] Available: http://www.digitalhumanities.org/Essays/ [Cited 23 November 2005].

Kroeze, J H 1996. The Applicability of Semantic Functions to Biblical Hebrew. *South African Journal of Linguistics* Supplement 33, 47-60.

Kroeze, J H 2003. The Semantic Functions of Embedded Constructions in Biblical Hebrew. *JNSL* 29(1), 107-120.

Lewis, D D & Jones, K S 1996. Natural Language Processing for Information Retrieval. *Association for Computing Machinery, Communications of the ACM,* 39(1), 92-101.

Petersen, U 2004. Emdros: A Text Database Engine for Analysed or Annotated Text. Paper Read at *20th International Conference on Computational Linguistics*, Geneva, 2004. [Online.] Available: http://emdros.org/petersen-emdros-COLING-2004.pdf [Cited 15 October 2004].

Ramsay, S *s.a.* Databases. *Essays in Humanities Computing.* [Online.] Available: http://www.digitalhumanities.org/Essays/ [Cited 23 November 2005]. (Reprinted from *A Companion to Digital Humanities.*)

Sperberg-McQueen, C M, Renear, A & Huitfeldt, C 2000. Meaning and Interpretation of Markup. *Markup Languages: Theory and Practice* 2(3), 215-234.

Thuraisingham, B M 2002. *XML Databases and the Semantic Web.* Boca Raton, FL: CRC Press.

Unsworth, J *s.a.* Knowledge Representation in Humanities Computing. *Essays in Humanities Computing.* [Online.]
Available: http://www.digitalhumanities.org/Essays/ [Cited 23 November 2005].

Wang, X & Dong, Y 2001. XML-based Data Cube. In *Proceedings of the Fifth International Conference on Info-Tech and Info-Net,* Beijing, China, 1 Nov. 2001, 48-53 (IEEE).

Wintner, S 2004. Hebrew Computational Linguistics: Past and Future. *Artificial Intelligence Review* 21(2), 113-138.

Witt, A 2005. Multiple Hierarchies: New Aspects of an Old Solution, in: Dipper, S, Götze, M & Stede, M (eds) 2005, *Heterogeneity in Focus: Creating and Using Linguistic Databases* (Interdisciplinary Studies on Information Structure 02), 55-85).