Deepening Prose Comprehension by Incremental Knowledge Augmentation From References

Amal Babour, Javed I. Khan, and Fatema Nafa
Department of Computer Science, Kent State University
Kent, Ohio, USA
Email: {ababour, javed, fnafa}@kent.edu

***Abstract*— Humans read references to gain a better understanding of a topic. In this paper, we propose a system that tries to mimic the human reading process for a given prose. The system can accommodate a deep prose comprehension by discovering the relevant parts from a reference related to the given prose that connect and illuminate a set of learnable concepts from the prose by adding direct meaningful knowledge paths among them. We present an evaluation model to measure the acquired knowledge and the learning process obtained by the system. The analysis of the results verifies that the system succeeded in deepening the prose comprehension.**

***Keywords*— Prose comprehension; Graph mining; Illuminated Semantic Graph; Knowledge paths; Sub Set Spanning.**

I. INTRODUCTION

Prose comprehension is an intriguing cognitive process [1]. Sophisticated prose is often rich with specialized concepts and terminologies that are sensitive and difficult for inexperienced readers to comprehend. This is observed in readings in many domains such as science and technology. Additionally, it is believed that the process of prose comprehension involves the integration of concepts with significant external knowledge, which is often called prior knowledge [2][3]. However, readers have different levels of prior knowledge, or sometimes they might not even have prior knowledge about a specific topic. Therefore, they need help through knowledge of full resources that allows them to compensate for the lack of prior knowledge [4]. However, the extensive number of references might have been a problem in itself. Readers might struggle to keep up with the type and the large amount of references, which can easily be disturbing. Additionally, searching for the relevant needed parts in the references is too extensive and time-consuming.

There is a great deal of work that tried to deepen understanding from prose by explicating the relationship among the text concepts [2][3][5], while there is another group of studies that employs external references to achieve deep comprehension [6][7][1]. The goal of this study is to present a method to develop our previous work [6]. In this paper, we present a method that reads the relevant parts from an external reference related to the given prose and discovers the direct knowledge paths connecting a set of learnable prose concepts. The main contributions of the paper are the following: First, we introduce an algorithm that reads the most appropriate parts from an external reference, such as Wikipedia, Encyclopedia, and textbooks and connects a set of learnable prose concepts by discovering the direct meaningful knowledge paths among them. Second, we present an evaluation model to be used by the system to measure the quantitative insight of the obtained knowledge and the learning process. Finally, we conduct three experiments on three texts of prose to assess and validate the effectiveness of the system.

The rest of the paper is structured as follows. Section II provides an overview of the related work. The main definitions and the overview of the system are presented in Section III. Section IV presents details of the used evaluation model. In Section V, we present the experiment and the evaluation results. The conclusion and the future work are presented in Section VI.

II. REALATED WORK

There has been several interesting studies on text comprehension. Some that focuses on *knowledge-dense* texts has highlighted deepening the understanding from the text itself, while others have focused on deepening the understanding using external consultation. Some of the most influential works on deepening text comprehension were introduced by Hardas and Khan. In [5], they posed the problem as a computational learning model in reading comprehension of natural texts that can mimic the growth of knowledge network as a step-by-step process of classification between recognized and unrecognized concepts during sentence-by-sentence reading. Later, using the computational model, they explored the impact of the concepts sequence on comprehension during reading [2]. Recently, Al Madi and Khan [3] developed the computational model to accommodate both text and multimedia comprehension. In the area of deepening the comprehension using external consultation, Babour and her associates addressed the problem of deepening text comprehension by bringing knowledge from more than one reference [7]. They proposed an automated method that iteratively selects a relevant reference to a given text that illuminates the text concepts by adding new knowledge paths using the selected relevant reference and ontology engine [6][7]. Later, they introduce a novel method that mines the appropriate parts from the relevant reference, which is valuable in deepening the comprehension by discovering the highest familiarity knowledge paths that connect a set of text concepts [1].

It would be relevant to discuss additional studies from graph mining perspective, which are relevant to the technique we have developed. Jin and his associates [8] proposed a graph-based retrieval model to detect a coherent chain between two given concepts across text documents. In [9],

 ISBN: 978-1-61208-518-0

Faloutsos and his associates developed a method that extracts a connected subgraph connecting two given nodes using electrical flow; whereas, Sozio and Gionis [10] proposed a method that extracts a compact subgraph of densely connected nodes by maximizing the minimum degree.

The work in this paper is about the same problem discussed in [1], but the difference is that our method is based on extracting the direct/shortest knowledge paths connecting a set of concepts instead of extracting the highest familiarity knowledge paths connecting them.

III. PROSE COMPREHENSION SYSTEM

The purpose of the system is to mimic the human reading process by creating an automated prose comprehension that discovers the hidden relations among each pair of concepts c_i and c_j in a learnable prose *LTX* and adds knowledge paths *K* among them using the learnable prose itself and a set of related references in an *Illuminated-Semantic-Graph G*.

We define the *Illuminated-Semantic-Graph G* as a graph *G=(C, E)* that provides a capture of the current state of the learning progress showing the learnable prose concepts C_L and the relationships between them found by reading the learnable prose *LTX*, the relevant parts from a related reference *RTX*, and the ontology engine *OE*, where *C* is a set of concepts $(c_1, c_2, \ldots, c_n)$ and *E* is a set of edges. The concept is either in *LTX*, *RTX*, or *OE* while the edge between any two concepts represents the relation between them. Each concept c_i can have one or more senses $(s_{i,1}, s_{i,2}, .. s_{i,x})$, where *i* is the concept number and *x* is the sense number. Each edge connects two concepts by a specific sense of each concept and has a label selected from *L'* representing the type of relation between the two concepts, where *L'* is a set of ontology engine and verb relations [1].

We define the *knowledge path K* as a path illuminating the relationship between two concepts, which can be represented as a sequence of edges that connects a concept c_i with a concept c_j in a preserved sense, where c_i and c_j are concepts from *LTX*. The in-between concepts in the path can be external to C_L. The type of the edge between any two concepts in the path is one of the following: Synonym, Hyponym, Hypernym, Meronym, Holonym, Instance or Verbed. The first six types are from the *OE*, and the last type is defined as the verb linked two concepts in the same sentence, where the two concepts are the subject and the object in the sentence [1].

Sometimes reading *LTX* only is not enough to understand, connect and illuminate the relation among the learnable concepts. Thus, there is a need to read a reference or set of references RTX_i to substitute the lack in the understanding. For example, given a specified *LTX* about 'Ethane' for comprehension and a list of five learnable concepts C_L= {ethane, hydrocarbon, hydrogen, gas, petroleum} in *LTX* as shown in Fig. 1 (A). The process of connecting C_L using different recources is shown in Fig. 1 (B).

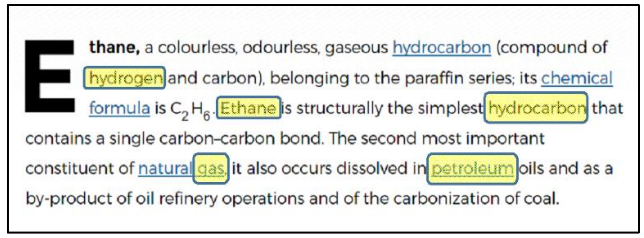
Ethane, a colourless, odourless, gaseous hydrocarbon (compound of hydrogen and carbon), belonging to the paraffin series; its chemical formula is C_2H_6. Ethane is structurally the simplest hydrocarbon that contains a single carbon-carbon bond. The second most important constituent of natural gas, it also occurs dissolved in petroleum oils and as a by-product of oil refinery operations and of the carbonization of coal.

Figure 1. (A) An example of LTX.

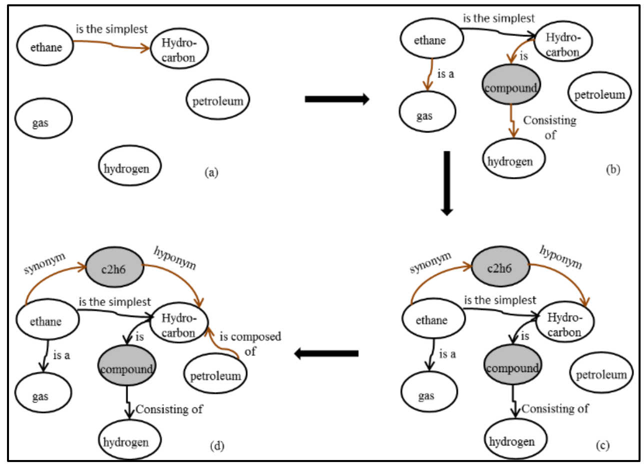

Figure 1. (B). The process of connecting C_L concepts using different resources. (a) Knowledge path K from LTX. (b) Knowledge path K using RTX1. (c) Knowledge path K using Ontology Engine OE. (d) Knowledge path K using RTX2.

Table I lists the symbols and definitions used, sorted by their overall appearance in the paper.

The overall system is applied on two core phases. The input of the first phase is the learnable prose *LTX* and C_L= $\{c_i, \ldots c_n\}$ in *LTX*. The system performs the ***Verbed-knowledge-paths*** $KP_v(\ )$ ***algorithm*** to generate an initial graph G_{LTX} ($G_{i=0}$) representing the verb relation between each pair of concepts in C_L, which is considered the output of this phase. The input of the second phase is a selected reference RTX_i related to *LTX* and C_L. The system performs the following algorithms in five steps each time it reads a new RTX_i.

1) ***Verbed-knowledge-paths*** $KP_v(\ )$ ***algorithm*** generates a graph G_{Ri} representing the verb relation between each pair of concepts in C_L from a RTX_i.

2) ***Sub-Set-Spanning algorithm SS( )*** extracts the M-sub-sets spanning paths from G_{Ri} that connect concepts from C_L with the direct meaningful knowledge paths. The extracted M-sub-sets are represented in G_{Ui} graph.

3) ***Merge algorithm Gmerge( )*** in the third step, generates G_{temp} that merges G_i and G_{Ui} graphs.

4) ***OE-knowledge-paths*** $KP_{OE}(\ )$ ***algorithm*** generates G_{Wi} graph representing the *OE* relation between each pair of concepts in G_{temp}.

5) ***Merge algorithm Gmerge( )*** in the fifth step, generates G_{i+1} that merges G_{temp} and G_{Wi}.

TABLE I. SYMBOLS AND DEFINITIONS

Symbol	Definition
LTX	The learnable prose.
G=(C, E)	Illuminated-Semantic-Graph.
$C_L = \{c_i, \ldots c_n\}$	A set of learnable noun concepts in the prose.
$RTX=\{RTX_1, RTX_2, \ldots RTX_n\}$	A set of reference texts.
OE	Ontology Engine.
C	A set of concepts.
$E = \{e_1, e_2, .., e_q\}$	A set of edges.
$s_{i,x}$	Is the x^{th} sense for concept c_i.
L'	a set of ontology engine and verb relations.
K	A sequence of edges constructing a Knowledge Path.
KPv( )	Verbed-knowledge-paths algorithm.
G_{LTX}/ G_0	The graph of the learnable prose.
G_{Ri}	A graph for a reference text.
SS( )	Sub-set-spanning algorithm.
G_{Ui}	The name of the graph extracted by SS( ).
Gmerg( )	Merge algorithm.
G_{temp}	Temporary graph.
KP_{OE}( )	OE-knowledge-paths algorithm.
G_{Wi}	The name of the graph created by KP_{OE}( )
G_{final}	The final graph generated after reading LTX and all RTX.
$v_{i,j}$	A verb connecting two concepts c_i and c_j in a sentence.
Υ	The maximum allowed distance between the concept and the verb in the verb relation in a sentence.
α	The maximum allowed length for K created by KP_{OE}( ).
β	Cluster Coefficient.
NIC_i	The neighbors interconnections coefficient of concept c_i.
deg_i	Degree of a concept c_i.
δ	Graph Entropy.
p_i	Probability of the concept c_i degree distribution.
$h_i(\Theta)$	Is the illuminated value for concept c_i at a particular phase.
Θ_i	Phase transition.
f_i	The frequency of concept c_i or the relation type extracted from Gutenberg corpus[14].
$H = \{h_i, h_j, \ldots\}$	Vector of Concepts Illumination Values {a quality between 0 and 1}.
$\|H\|$	Is the summation of h_i for each c_i in C_L.
$a_{i,j}$	An element denoting the association strength between concept c_i and c_j.
A	A matrix with $a_{i,j}$ elements.
$\tilde{N}$	The number of connected concepts.

After reading the whole set of RTX_i, the system generates the G_{final} that includes a set of *K*, where both ends of each *K* are from the C_L.

Fig. 2 explains the phases of the system in detail. The bold line in phase 2 shows the iterative process of applying the proposed algorithm with each reading of a new RTX_i for finding the direct meaningful knowledge path among C_L. Both *LTX* and *RTX* go through preprocessing. During preprocessing, all stopwords, except negation words, are removed and the remaining words are stemmed using Porter Stemmer [11]. The next section describes each algorithm in detail.

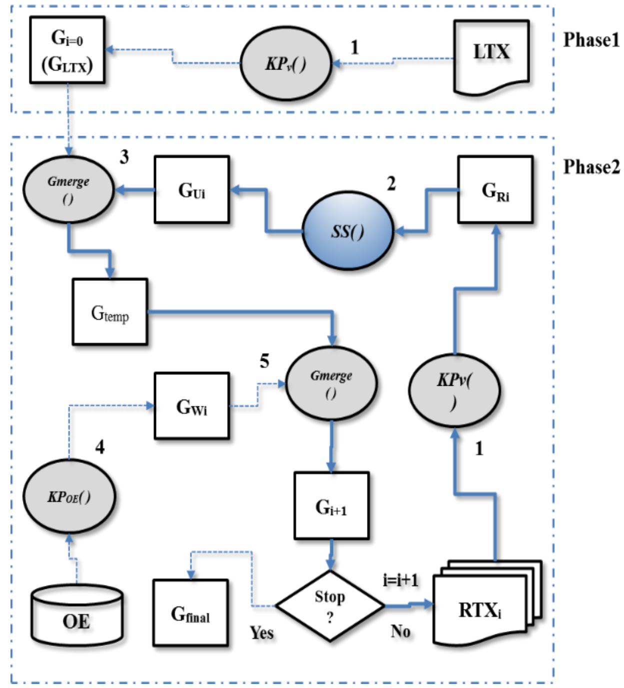

Figure 2. Overview of the system.

A. *Verbed-Knowledge-Paths algorithm KP_v( )*

Given a *LTX* or RTX_i and a C_L, for each sentence *in LTX* or RTX_i, the algorithm searches for any pair of concepts (c_i, c_j) from C_L to see if there is a verb $v_{i,j}$ between them, where the distance between c_i and $v_{i,j}$ and the distance between $v_{i,j}$ and c_j is less than or equal a threshold Υ. If so, it saves them in the form of $[c_i, v_{i,j}, c_j]$ as an edge in the graph representing a verb relation between a pair of concepts c_i and c_j. If $v_{i,j}$ is preceded or followed by a negative word, the negative word is attached to the verb forming one word. The output of the algorithm is a graph that represents the verbed relation between any pair of concepts from C_L.

B. *Sub-Set-Spanning algorithm SS( )*

The algorithm in Fig. 3 represents the Sub-Set-Spanning algorithm as follows: The input of the algorithm is G_{Ri} and C_L, where the output is G_{Ui}, which is a subgraph from G_{Ri} that presents the direct paths among C_L. We use the same algorithm used in our previous work [1], but we replace the highest familiarity knowledge paths among the concepts with the direct ones.

The search for a direct knowledge path has been implemented as a breadth-first-search (BFS). For each component *comp* in *G*, the algorithm uses a queue data structure *Queue* to temporarily hold each visited concept in the graph with its neighbors. It picks any concept from C_L as the source *s* for initializing the *Queue*. Then, it initializes the distance *dist* between *s* and each concept *c* in the *comp* to *INFINITY* and initializes the previous concept *prev* of each *c* to *-1*. In the loop iteration, it de-queues the first concept *c* in the queue, marks it as visited, and checks if $c \in C_L$. If so, it updates its *dist* to *0*, adds it to *M* where *M* holds the found C_L

concepts and removes it from C_L. Then, it en-queues all the neighbors c_i*'s* of concept *c* if they are marked as non-visited, assigns *prev* and calculates *dist* for each of them. If the current *dist* of c_i is less than its previous *dist*, that means a shorter knowledge path to c_i is found. The c_i*'s prev* and *dist* are updated to the new less values and the process is repeated till the queue becomes empty. If all *comp* are checked, *getPaths* constructs the M sub-sets spanning from *M* and *prev*. The returned M-sub-sets spanning are represented in G_{Ui}.

Fig. 4 shows an example of the M sub-sets spanning returned by SS( ) algorithm, where C_L ={'ethane', 'carbon', 'petroleum'}. The returned M sub-set spanning is {['ethane', 'chemical', 'carbon', 'constituent', 'petroleum']}.

```
Def Sub-Set-Spanning ( ):
Input: GRi, , CL
Output: M-sub-sets spanning.
// initialization
for each comp in G:
  Queue=ϕ
  s= pick any member from CL
  enqueue(Queue,s)
  if CL ≠ ϕ :
    for each concept c in comp
      prev[c]=-1
      dist[c]=INFINITY
      Visited[c]=False
  While Queue ≠ ϕ:
     c= dequeue(Queue)
    Visited[c]=True
    if c in CL:
      dist[c]= 0
       add c to M
       remove c from CL
    for each neighbor  ci of c:
      if ci not in Queue and Visited[ci]==False:
        enqueue(Queue,ci)
      alt= dist[c]+ 1
      if alt < dist[ci]
        prev[ci]=c
        // a shorter knowledge path to ci has been found
        dist[ci]=alt
M-sub-sets = getPaths(M[ ], prev[ ])
return M-sub-sets
```

Figure 3. Sub-Set-Spanning algorithm.

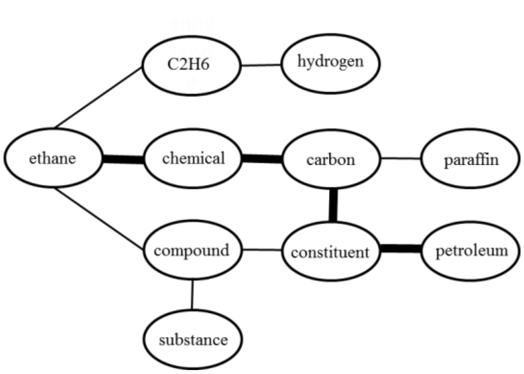

Figure 4. M Sub-Set-Spanning example.

C. Merge algorithm Gmerge( )

The algorithm merges two graphs into a single one.

D. OE-knowledge-paths algorithm KP_{OE}( )

The algorithm searches for knowledge paths *K* of a length less than or equal to threshold α connecting each pair of concepts that appear in G_{temp} if found using an ontology engine. The algorithm is presented in detail in our previous work [6].

IV. SYSTEM EVALUATION MODEL

In this section, we present a set of measurements, which are employed to assess the quantitative knowledge gained from *G*, including information content, graph organization, richness of information, concept illumination value, and knowledge paths.

A. Information content

The size of the graph is measured by the whole number of concepts *C* and the associations *E* among them, where the concepts belong to three different sources *LTX*, *RTX*, and *OE*. High size is a good indicator to a wealth of information and therefore deep comprehension. The process of prose comprehension is completed by reading the last RTX_i in which the graph transforms from (G_0, G_1,……,G_{final}). Therefore, the size of *G* is increased and the information is grown respectively.

B. Graph organization quality

The graph organization plays an important role in predicting the performance of the learning progress. A good graph organization gives a clarification about the context of each concept and how each concept is related to other concepts by representing groups of strongly connected concepts each works as constrains on the possible meaning of its concepts, therefore the meaning of the concepts can be greatly clarified. It can be measured by clustering coefficient β, which offers a way to measure how the concepts in the graph tend to form groups of strongly connected concepts. According to [12], we suggest calculating β using (1); the closer to 1 value indicates the higher clustered graph.

$$\beta = \sum_{i=0}^{n} \frac{2NIC_i}{deg_i(deg_i - 1)} \tag{1}$$

C. Richness of Information

Information richness is a measure of how much information a graph contains. High information richness usually indicates a graph rich with information and deep comprehension. It can be measured by entropy δ, which measures the amount of information within the graph. According to [13], we calculate δ using (2):

$$\delta = -\sum_{i=0}^{n} p_i log(p_i) \tag{2}$$

Where p_i is determined by (3):

$$p_i = \frac{deg_i}{2|E|} \tag{3}$$

D. Calculating the concepts illumination values H

The concept illumination value h_i is a way to interpret the level of understanding the concept. It presents the importance of the concepts at each particular phase. The higher the concept illumination value, the more understanding there is in the prose. The initial illumination value of a concept can by calculated using (4). This initial value represents the prior knowledge or the familiarity of the concept, where *h(0)* represents the initial value of concept *i*. The high frequency means the high familiarity of the concept.

$$h_i(0) = -1/\,log\left(\frac{f_i}{10^9}\right) \tag{4}$$

Tracking the growth evolution of the concept illumination value during the learning progress is an interesting approach to measure the deepening of prose comprehension. We calculate the illumination value of each concept at each phase. We consider the phase Θ_i as reading a set of sentences. Then, we estimate how the illumination value varies over the learning process through a set of phases. After a set of phases, the concept illumination value reaches a stable value which is considered its final illuminated value. The learning progress at each phase is assessed by the value of $|H|$ which is the summation of h_i for each c_i in C_L. The higher the $|H|$, the deeper the learning. To calculate h_i for each concept in the graph at each phase, we utilize (5).

$$H(\Theta+1) = transpose(A) * H(\Theta) \tag{5}$$

$$\overbrace{\begin{array}{c|ccccc} & c_i & & & & c_n \\ \hline c_i & a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,n} \\ .. & \cdots & \cdots & \cdots & \cdots & \cdots \\ .. & \cdots & \cdots & \cdots & \cdots & \cdots \\ .. & \cdots & \cdots & \cdots & \cdots & .. \\ c_n & \cdots & \cdots & \cdots & \cdots & a_{n,n} \end{array}}^{A} * \overbrace{\begin{bmatrix} h_1 \\ h_2 \\ \cdots \\ \cdots \\ h_n \end{bmatrix}}^{H,\ h_i}$$

We will consider the association strength $a_{i,j}$ as the illumination value of the relation type between a pair of concepts (c_i, c_j). The value of $a_{i,j}$ is calculated by (4), f_i here represents the frequency of the relation type extracted from Gutenberg corpus [14], where high frequency means high familiarity of the relation type. The relation between *f* and *h* is a direct relation. This means the higher the frequency, the higher its illumination value. Table II shows different types of relations, which are common between any pair of concepts.

TABLE II. RELATION STRUCTURE BETWEEN ANY PAIR OF CONCEPTS

Relation type	Relation structure	$a_{i,j}$ value
verb relation	**Case#1: *single verb:*** c_i – : $s_{i,*}$ – v1 – $s_{j,*}$: – c_j	$h_{v1}(\Theta)$
	Case#2: *dual verb:* c_i – : $s_{i,*}$ – v1 v2 – $s_{j,*}$: – c_j	$h_{v1}(\Theta) * h_{v2}(\Theta)$
	Case#3: *dual paths:* c_i – : $s_{i,*}$ – v1 v2 – $s_{j,*}$: – c_j c_i – : $s_{i,*}$ – v3 v4 – $s_{j,*}$: – c_j	$h_{v1}(\Theta) * h_{v2}(\Theta)$ + $h_{v3}(\Theta) * h_{v4}(\Theta)$
Wordnet relation	**Case#1: *Class/sub-class:*** c_i – : $s_{i,*}$ – Hypernym – $s_{j,*}$: – c_j or c_i – : $s_{i,*}$ – Hyponym – $s_{j,*}$: – c_j	$h_{class}(\Theta)$
	Case#2: *Part/sub-part:* c_i – : $s_{i,*}$ – Holonym – $s_{j,*}$: – c_j or c_i – : $s_{i,*}$ – Meronym – $s_{j,*}$: – c_j	$h_{part}(\Theta)$
	Case#3: *synonym:* c_i – : $s_{i,*}$ – Synonym – $s_{j,*}$: – c_j	$h_{synonym}(\Theta) = 1$

E. Types of Knowledge Paths

The illumination-semantic-graph is a complex graph of concepts and associations. The graph has many interconnected concepts, ultimately leading to a congested graph. Hence, the information becomes hard to read; for example, it is hard to trace a particular sequence of edges connecting two concepts because the edges overlap. This can be clarified by extracting knowledge paths. A knowledge path is a way to reveal underlying information in the graph tidily. For more clarification, we classified the knowledge paths into seven types described in Table III.

TABLE III. KNOWLEDGE PATHS TYPES

	K types	Description
1.	Genesis-Set	Where each label in the sequence of edges of *K* has either a hyponym or a hypernym relation.
2.	Synonym-Set	Where each label in the sequence of edges of *K* has a synonym relation.
3.	Part-of-Set	Where each label in the sequence of edges of *K* has either a meronym or a holonym relation.
4.	Conceptual-Neighbor-Set	Where the labels in *K* have a combination of hyponym and hypernym relations.
5.	Structural-Neighbor-Set	Where the labels in *K* have a combination of meronym and holonym relations.
6.	Complex-Neighbor-Set	Where the labels in *K* have a combination of hyponym or hypernym and meronym and holonym relations.
7.	Verbed-Set	Where each label in the sequence of edges of *K* has a verb relation.

V. EXPERIMENT AND EVALUATION

In this section, we evaluate the proposed system based on the statistical characteristics of the obtained graphs of three experiments, which indicate the quantitative insight of the amount of comprehension that can be gained by the readers. In the future work, we are going to perform the experiments with actual readers. The selected proses LTX_i used in the experiments, as well as the C_L for each are shown in Table IV.

 ISBN: 978-1-61208-518-0

TABLE IV. LIST OF THE PROSES USDED IN THE EXPEIEMENTS

	LTX	C_L
Experiment1	LTX1: 'Ethane chemical compound' [15]	['Ethane', 'hydrocarbon', 'hydrogen', 'carbon', 'carbon-carbon', 'petroleum', 'carbonization', 'coal']
Experiment2	LTX2: 'New Test for Zika OKed' [16]	['zika', 'infection', 'dengue', 'hikungunya', 'virus', 'aedes', 'mosquito', 'antibody']
Experiment3	LTX3: 'Anesthesia gases are warming the planet' [17]	['Anesthetic', 'carbon', 'climate', 'oxide', 'desflurane', 'isoflurane', 'sevoflurane', 'halothane']

The used *OE* is Wordnet [18] version 1.7 and the used *RTX* is Wikipedia. For each experiment, *RTX* is a set of articles selected from Wikipedia about each concept in C_L. We applied the automated method used in [7] for the selection of the Wikipedia articles. For each experiment, the system goes through eight RTX_i and creates nine *G*, G_0 represents the relation among C_L in *LTX* and eight G_i each represents the relation among the C_L after adding reading a new RTX_i.

A. Graph Analysis

In this section, we present our analysis of the information gained from *G*. The breakdown of the total number of concepts *C* and the number of edges *E* in G_0 and G_{final} are shown in Table V, where the concepts are from *LTX*, *RTX*, and/or *OE*. It is observed that there is a variance in the number of concepts and edges between G_0 and the G_{final}, which is a good indicator to the plentiful information in the G_{final}, hence the depth of prose comprehension.

TABLE V. BREAK DOWN OF THE TOTAL NUMBER OF EDGES AND CONCEPTS IN THE FINAL G

	Experiment1		Experiment2		Experiment3	
	G_0	G_{final}	G_0	G_{final}	G_0	G_{final}
E	3	100	1	76	0	29
Number of *LTX* concepts	8	8	8	8	8	8
Number of *RTX* concepts	0	7	0	8	0	4
Number of *OE* concepts	0	36	0	22	0	9

Furthermore, Fig. 5 shows the number of connected learnable prose concepts C_L in G_i, where (x-axis) refers to the G_i after adding each RTX_i and (y-axis) is the number of connected concepts per G_i. For each experiment, we can observe that the number of connected concepts $\tilde{N}$ is increased when the system reads RTX_i. The concepts become fully connected after reading the 8th *RTX, 2nd RTX, and 1st RTX* for *LTX1*, *LTX2, and LTX3* consecutively, which verifies the effectiveness of the system for connecting C_L.

Fig. 6 shows the clustering coefficient β observed in each G_i, where (x-axis) is the G_i and (y-axis) is the clustering coefficient β. It is obvious that some of the graphs especially for the first experiment are highly clustered, which signifies that their concepts are highly clustered together.

B. Knowledge Analysis

In this section, we present our analysis of the learning progress on *LTX* comprehension from *G* in the three experiments. Fig. 7 represents the entropy δ per each G_i, where (x-axis) is the G_i and (y-axis) is the entropy δ. It is observed that the δ in the three experiments starts with a low value, then it increases gradually after reading a new RTX_i, which indicates that the graph concepts become more influential each time the system reads a RTX_i.

Moreover, Fig. 8 plots the variance in the concepts illumination values $|H|$ (y-axis) of C_L with the phases of learning progress Θ_i (x-axis) in the G_{final}. We examined 50 phases. We can clearly see from the plot that $|H|$ increases gradually over the phases especially in the first experiment, which indicates the deeper comprehension of the C_L and the *LTX* after each phase Θ_i.

C. Knowledge Paths Classification

The breakdown of *K* types that are found in G_{final} are shown in Table VI.

TABLE VI. BREAKDOWN OF KNOWLEDGE PATHS TYPES

	Experiment 1	Experiment 2	Experiment 3
Genesis-Set	2	0	2
Synonym-Set	0	0	0
Part-of-Set	0	0	0
Conceptual-Neighbor-Set	6	0	2
Structural-Neighbor-Set	0	0	0
Complex-Neighbor-Set	0	0	0
Verbed-Set	17	26	8

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a computerized human prose comprehension system that discovers relevant parts from a reference that connect and illuminate the learnable concepts by direct meaningful knowledge paths among them. The system is an improved version of our previous work [6]. The statistical results obtained from the graph(s) show that the system succeeds in connecting the learnable concepts by discovering the direct meaningful knowledge paths among them and in achieving a deep prose comprehension. For future work, we are going to compare the results of the used method with the one discussed in [1]. We are also going to test the impact of the system results on the comprehension of actual readers.

REFERENCES

[1] A. Babour, J. I. Khan, and F. Nafa " Deepening Prose Comprehension by Incremental Free text Conceptual Graph Mining and Knowledge," , Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), International Conference on. IEEE, 2016 (in press).

[2] I. Khan and M. S. Hardas, "Does sequence of presentation matter in reading comprehension? A model based analysis of semantic concept network growth during reading," IEEE, 2013, pp. 444–452.

[3] N. S. Al Madi and J. I. Khan, "Is learning by reading a book better than watching a movie? A computational analysis of semantic concept network growth during text and multimedia comprehension," IEEE, 2015, pp. 1–8.

[4] J. E. Moravcsik and W. Kintsch, "Writing quality, reading skills, and domain knowledge as factors in text comprehension," Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale, vol. 47, no. 2, 1993, pp. 360–374.

[5] M. Hardas and J. Khan, "Concept learning in text comprehension," in Lecture Notes in Computer Science. Springer Science + Business Media, 2010, pp. 240–251.

[6] A. Babour, F. Nafa, and J. Khan, "An Iterative Method for Enhancing Text Comprehension by Automatic Reading of References," in ThinkMind(TM) digital library, 2015, pp. 66–73.

[7] A. Babour, F. Nafa, and J. I. Khan, "Connecting the dots in a concept space by Iterative reading of Freetext references with Wordnet," vol. 1, IEEE, 2015, pp. 441–444.

[8] W. Jin, R. K. Srihari, and X. Wu, "Mining concept associations for knowledge discovery through concept chain queries," in Advances in Knowledge Discovery and Data Mining. Springer Science + Business Media, 2007, pp. 555–562.

[9] C. Faloutsos, K. S. McCurley, and A. Tomkins, "Fast discovery of connection subgraphs," ACM, 2004, pp. 118–127.

[10] M. Sozio and A. Gionis, "The community-search problem and how to plan a successful cocktail party," ACM, 2010, pp. 939–948.

[11] M. F. Porter, "An algorithm for suffix stripping," Program: electronic library and information systems, vol. 14, no. 3, 1980, pp. 130–137.

[12] D. G. Bonchev and D. H. Rouvray, "Quantitative measures of network complexity." in Complexity in chemistry, biology, and ecology, Springer US, 2005, pp. 191-235.

[13] R. Navigli and M. Lapata, "Graph Connectivity Measures for Unsupervised Word Sense Disambiguation," IJCAI, 2007, pp. 1683-1688.

[14] M. Hart. Project Gutenberg. 1971.

[15] The Editors of Encyclopædia Britannica. (2016, September 22). Ethane |chemical compound. Retrived from: https://www.britannica.com/science/ethane .

[16] K. Grens. (2016, March 22). New Test for Zika OKed. Reterived from: http://www.the-scientist.com/?articles.view/articleNo/45638/title/New-Test-for-Zika-OKed .

[17] E. DeMarco. (2015, April 7). Anesthesia gases are warming the planet. Retrieved from: http://www.sciencemag.org/news/2015/04/anesthesia-gases-are-warming-planet .

[18] Princeton, "About WordNet - WordNet - about WordNet," Trustees of Princeton University, 2016.

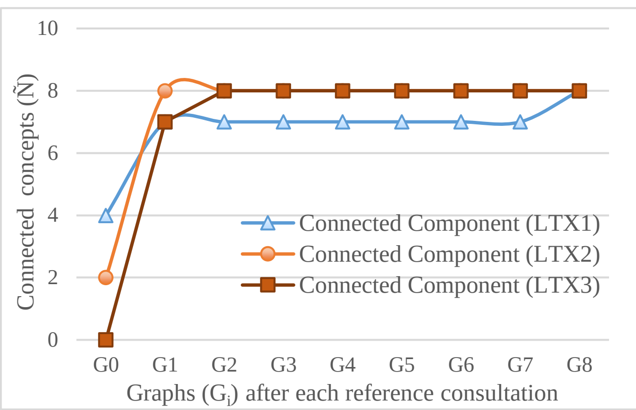

Figure 5. Learnable Prose Concepts connectivity per graphs G_i.

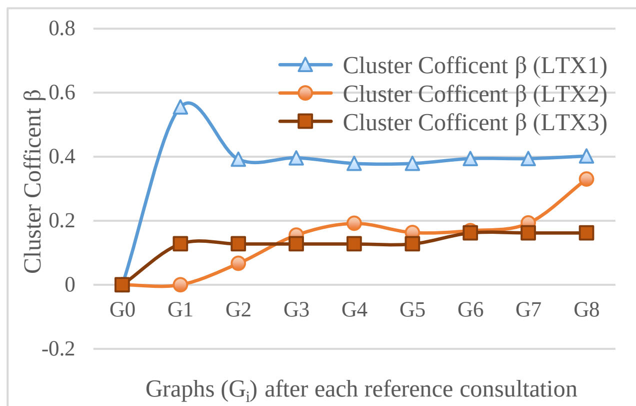

Figure 6. Cluster Coefficient per graphs G_i.

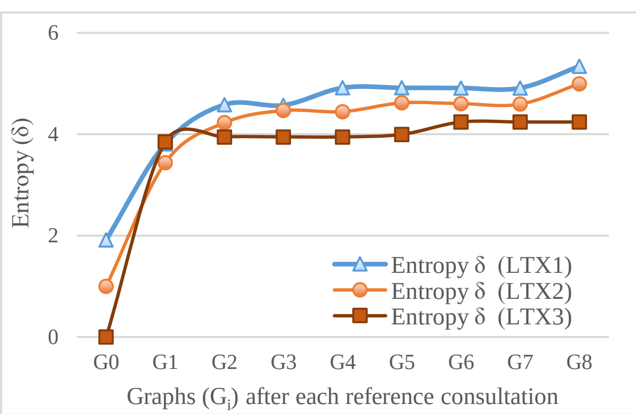

Figure 7. Entropy per graphs G_i.

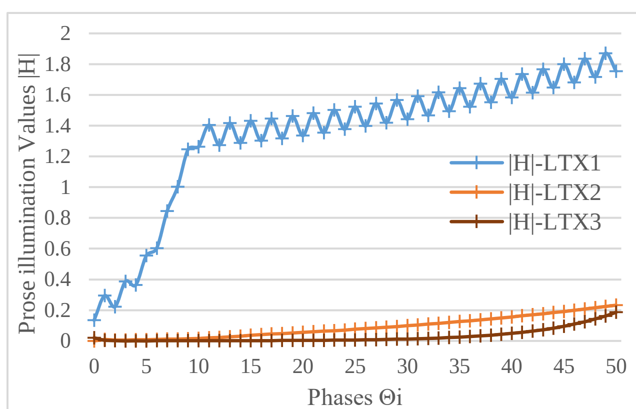

Figure 8. Prose illustration values per phases.