An Iterative Method for Enhancing Text Comprehension by Automatic Reading of References

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

***Abstract*— Humans read references to increase understanding about a topic. In this research, we investigate an interesting algorithm which tries to mimic the human reading process. Given a free text about a topic, the algorithm iteratively discovers important references to illuminate the less understood part of the text at hand and then analyzes the reference text to add new knowledge paths. The algorithm also consults modern semantic dictionary through the analysis as needed. In this paper, we are going to share an experiment, which uses Wikipedia pages as reference and WordNet as the ontology engine to understand a news article. We display the knowledge gain via this algorithm.**

Keywords- WordNet; Wikipedia; Semantic-Graph; Illuiminated-Semantic-Graph.

I. INTRODUCTION

Some texts are very complicated and have very specialized concepts that are difficult to comprehend. This happens in many domains, such as science, politics, and language. Text comprehension requires not only the extraction of individual concepts in the text, but also the identification of the relations among the concepts [1]. Text comprehension can be defined as a cognitive process of understanding and comprehending concepts from a text and the relationship among them. The more relations discovered, the better the comprehension of the text. Sometimes the relation between two concepts is explicit, while the relation between two other concepts is implicit and needs to use external reference texts to extract it. An ontology engine is applied to find the relation directly or by detecting a path between the concepts. In some cases, using ontology engine alone does not help. In this case, using additional reference texts can help in finding the hidden relations among the text concepts and adding new knowledge. Most of the recent techniques in text understanding are based on ontology engines, reference texts or both together to find the relations among the text concepts. Although anyone who wants to get a deep comprehension about a specific text back to external references to read about it, the recent techniques do not use the reference texts themselves, but utilize their categories instead. On the other hand, the use of the ontology engines alone gives a short connection. Therefore, these techniques are considered very limited. In the proposed system, we mimic the human process of reading by making a computerized text comprehension to discover the most relevant references that illuminate the relation among the text concepts and add new knowledge about them by an automatic reading of the important reference texts to make the text comprehension easier for the reader.

The technique presented in this research falls in the category of free text based on ontology engine and the reference texts themselves. We demonstrate the way by which the system gives a deep text comprehension, by an iterative process through different reference texts. Thus, the information about the concepts relations is brought from more than one reference. This makes humans have a faster and wider knowledge. The singular value decomposition (SVD) algorithm is applied to extract the most important concepts in the given text and the references. This system can be used to enhance a lot of models, such as Toplet, which is a deep structure of a specific topic [16].

The rest of the paper is structured as follows. Section II provides an overview of the related work. The main definitions and the system are presented in Section III. In Section IV, we present detailed experimental results demonstrating dramatic improvement in the extracted relations among the concepts after adding the external reference texts. Section V presents the conclusion and the future work.

II. RELATED WORK

There have been interesting researches using an ontology engine such as WordNet to extract relations among concepts. Kang and his associates [1] used Guided Agglomerative Hierarchical Clustering algorithm (GAHC) [2] to identify the semantic relationship among concepts in order to construct concepts of multi-branch hierarchy of a domain specific text corpus automatically. In addition to the “is-a” and “part-of” relations from WordNet, they incorporate a set of lexico-syntactic patterns extended from Hirst’s [3] to identify “is-a” relations from a domain specific corpus, and applied related lexico-syntactic patterns [4] to extract “part-of” relations. Yadav and his associates [5] construct a semantic graph from a text document, applying WordNet to find the direct relation among the concepts automatically to be used for different applications in text mining such as keyword extraction and knowing the nature of the document. Tu and his associates [6] proposed an approach based on text concepts and WordNet to build a multi-way concept hierarchy from a document corpus. They applied multi-way agglomerative clustering algorithm [7] to generate the concept hierarchy automatically. On the other hand, a number of researches applied reference texts such as Wikipedia to extract the concept relations. Taieb and his associates [8] presented the idea of using information content (IC) metric based on Wikipedia Category graph [8] as semantic taxonomic resource for extracting relations between concepts. Lipczak and his associates [9] utilized

Wikipedia category graph to propose a system that spots mentions of entities in a document and link the mentions to corresponding Freebase articles. Han and Mon [10] use the taxonomy of categories in Wikipedia to compute semantic relations using structured knowledge extracted from Wikipedia. They introduce strategy over the network of Wikipedia categories to evaluate semantic relations.

III. Overview of the System

The purpose of our system is to create *Illuminated-Semantic-Graph* which represents hidden relations among each pair of concepts in a target text *T*, where *T* is the input to the system and the *Illuminated-Semantic-Graph* is the output. In this paper, the term concept refers to a particular sense of noun.

The *Illuminated-Semantic-Graph* is a directed graph G=(V,E) where V is a set of concepts and E is a set of edges. Each concept v_n can have one or more senses (s_{n1},s_{n2},..s_{nk}) where n is the concept number and k is the sense number. Each edge connects two concepts via a specific sense of each concept. A label represents the type of relation between any two concepts is shown over each edge. The concept is either in T, ontology engine, or reference text, while the edge between any two concepts represents the relation between the concepts via the following relation types (synonym (S), hyponym (H), meronym (M), and instance (I)). We do not mention the hypernym and holonym types because they are the inverse relations of hyponym and meronym respectively. Since we are dealing with noun concepts, we focused only on the mentioned relations .We define a *Knowledge Path* $P_{i,j}$ as a path which preserves the sense in G. It is a sequence of edges that connects a concept C_i with a concept C_j.

Figure 1 shows two examples of geometric paths: C_1 - $\{S_{11}\text{-}I\text{-}S_{21}\}$ - C_2- $\{S_{21}\text{-}S\text{-}S_{31}\}$ -C_3- $\{S_{31}$ -$I\text{-}S_{51}\}$ - C_5 and C_1 - $\{S_{11}\text{-}I\text{-}S_{21}\}$ - C_2- $\{S_{21}\text{-}S\text{-}S_{31}\}$ -C_3- $\{S_{32}\text{-}H\text{-}S_{41}\}$ – C_4 – $\{S_{41}\text{-}H\text{-}S_{64}\}$-$C_6$, where C_1, C_2 … etc. refer to the concept number. S_{11} refers to the first sense of the first concept, S_{21} the first sense of the second concept, etc. The Knowledge paths can be extracted from the geometric-paths. For example, C_1 - $\{S_{11}\text{-}I\text{-}S_{21}\}$ - C_2- $\{S_{21}\text{-}S\text{-}S_{31}\}$ -C_3- $\{S_{31}$ -$I\text{-}S_{51}\}$ - C_5 and C_3- $\{S_{32}\text{-}H\text{-}S_{41}\}$ – C_4 – $\{S_{41}\text{-}H\text{-}S_{64}\}$-$C_6$ are considered knowledge paths because the incoming and the outgoing senses for each concept is preserved. However, not all the knowledge paths are correct. Some of them are classified as noise paths. For example, C_1 is an instance of C_2, C_5 is instance of C_3, and C_2 and C_3 are synonyms, so C_1 and C_3 seem to be equivalent, while they are not. So C_1 - $\{S_{11}\text{-}I\text{-}S_{21}\}$ - C_2- $\{S_{21}\text{-}S\text{-}S_{31}\}$ -C_3- $\{S_{31}$ -$I\text{-}S_{51}\}$ - C_5 is considered a noise path.

The idea of the system is represented in six steps:

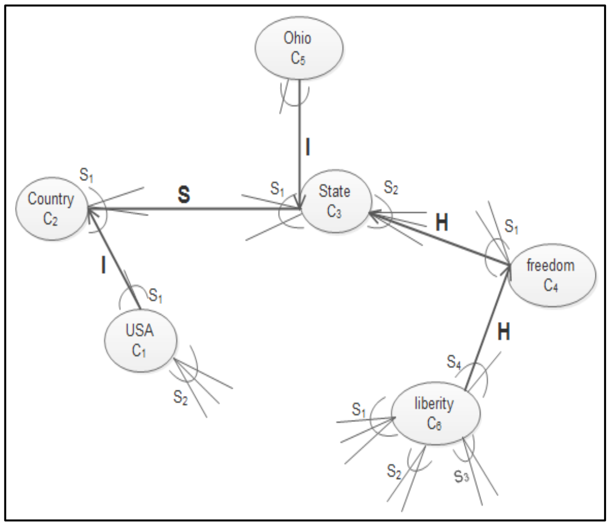

Figure. 1. Geometric-paths examples.

1. Extracting the most important concepts in *T* to create the *Target-Text-graph.*
2. Using an ontology engine to discover the relation between each pair of concepts extracted in step 1 by adding knowledge paths between each pair if found. A number of ontology engine concepts are added to the graph. The graph in this version is called *Semantic-graph*.
3. Extracting the most important concepts from a reference text *i*, which is a text related to *T.*
4. Repeat step 2 on the important concepts extracted in step 3. The graph in this version is called *Reference-Sematic-Graph.*
5. Add more knowledge paths from the *Reference-Semantic-Graph* created in step 4 to the *Semantic-Graph* created in step 2, while either of the added knowledge paths ends is a *T* concept and the other end is a new concept from the reference text concepts. A number of ontology engine and reference text concepts are added to the graph. The graph in this version is called *Illuminated-Semantic-Graph*.
6. Repeat 3-5 steps for the reference text *i++*.

Figure 2 explains the steps.

Figure 3 illustrates an example that shows how adding more knowledge paths from the *Reference-Sematic-Graph* to the *Semantic-Graph* gives more text comprehension. Some of the knowledge paths give concept-illuminations by adding new knowledge to *T* concepts, while other knowledge paths give relation-illuminations by showing additional hidden relations between the knowledge paths ends of *T* concepts.

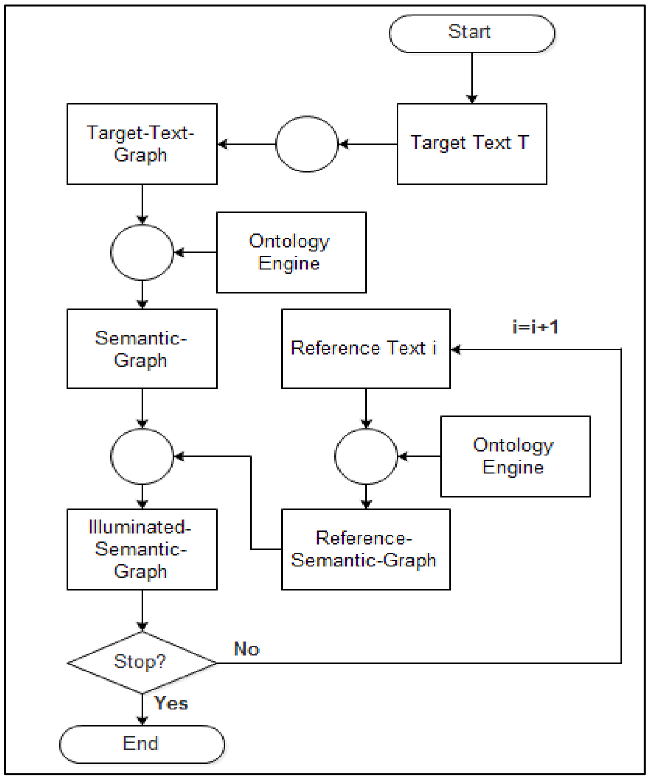

Figure 2. Workflow to generate the Illuminated-Semantic-Graph.

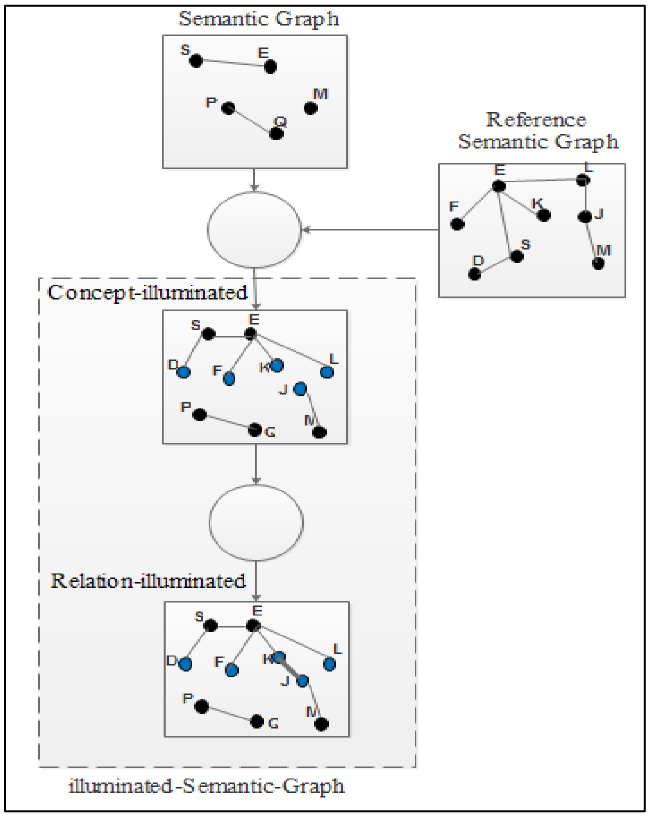

Figure 3. Example of Concept illumination and Relation Illumination.

The overall system uses two core techniques (i) **concepts extraction** and (ii) **knowledge paths detection**. Both techniques are applied on two phases **Target Text phase** and **Reference Text phase**, where the input of the Target Text phase is *T* and the output is the *Semantic-graph*. The *Target Text graph* is generated after applying the first technique in this phase, where the SVD algorithm is used to generate n` which is the most important concepts in *T*. The input of the Reference Text phase is a text R_T from any reference such as Wikipedia. This text is selected based on n_c concepts where n_c is a list of *T* concepts that have knowledge paths among them in the *Semantic-Graph*, $n_c \subseteq$ n`. The output of the Reference Text phase is a *Reference-Semantic-Graph*. A number of knowledge paths generated from *Reference-Semantic-Graph* are added to the *Semantic-Graph* to generate the *Illuminated-Semantic-Graph* where either of the added knowledge paths ends is a *T* concept and the other end is a new concept from R_T. The more added knowledge paths, result in more concepts and relations illumination for *T* concepts.

Figure 4 explains the process in details. *T and* R_T go through various text preprocessing. They are pruned by removing stop words and stemming words using Porter stemmer algorithm.

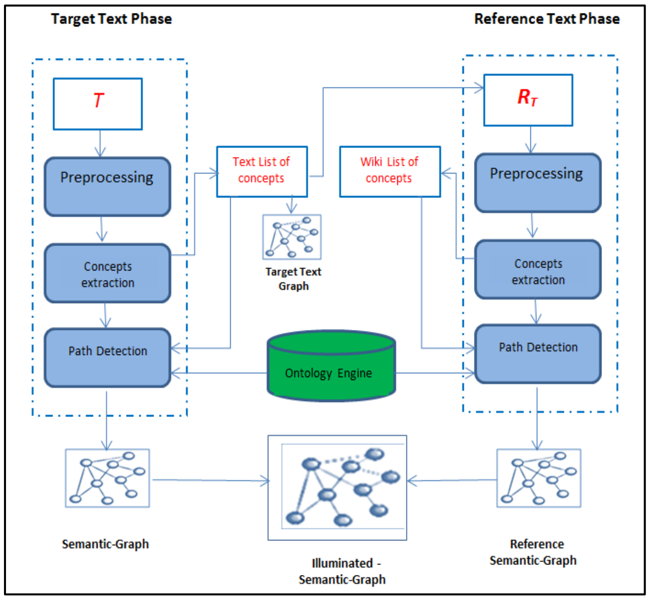

Figure 4. Overview of the System.

The **concepts extraction** technique generates a list of the most important concepts in *T* based on SVD method. **Knowledge path detection** technique finds the knowledge paths between each pair of concepts if found using an ontology engine.

A. *Concepts extraction:*

This technique uses SVD [11] to extract the most important concepts in the text using the following equation:

$$A=USV^T$$

Where *A* is n-by-n matrix and n represents all concepts/nouns in the text. Each cell in *A* represents the number of appearance of pair of nouns in each sentence in the text. *U* is an orthogonal matrix, *S* is a diagonal matrix, and *V* is the transpose of an orthogonal matrix. The SVD is used to reduce the high dimensionality of matrix *U*, emphasize the important concepts in T, and eliminate any noise. The first dimension of *U* is ordered in descending order to get the n` of the most important concepts in the T. A number of pairs between concepts are generated from n`. Table I gives an example of concept pairs.

TABLE I. CONCEPTS PAIRS DATA STRUCTURE

i	**Concept pairs**
1	(judge, charge)
2	(Egypt, Israel)
3	(case, family)
..	..

B. *Knowledge paths detection:*

Function *discover-knowledge-paths* in Figure 5 searches for knowledge paths of length less than or equal *k* connecting each pair of concepts/nodes if found using an ontology engine. Its input is Sword and Eword, where Sword and Eword are respectively the first and last nodes in the path. R is a dictionary of all relations between concepts in the ontology engine, and relationalGraph is a dictionary used to hold concepts that have any type of relations from R with the last node in the current path. The search for a knowledge path has been implemented as a breadth-first search (BFS). The function searches in all senses of Sword. For each sense, it searches for knowledge paths from Sword node to Eword node by searching the neighbors of the Sword that have any type of relations from R and have the same sense of Sword. Then, it searches the neighbors of the neighbors, and so on until it reaches Eword. The function returns the knowledge paths of length less than or equal *k* connecting Sword and Eword. If the knowledge path does not exist, the function returns ‘Not Found’. The function does not return the shortest path between the pair of concepts as the knowledge path because it could be a path of multi senses concepts.

The function uses two queue data structures. One is NodeQ and the other one is PathQ. The NodeQ saves the current path in the BFS that has the node to explore the next. The PathQ holds the created paths until now. The function starts with Sword as the current path. The while loop iterates through the paths in PathQ searching for a knowledge path connects Sword and Eword. In each loop iteration, it de-queues the first path in PathQ and signed it in NodeQ. Then, it checks if the last node in NodeQ matches Eword. If so, the function saves the knowledge path in Kpaths. If not, it checks if the length of the NodeQ is less than k, if so, for the sense of the last node in NodeQ, the function gets all the concepts that have one of the relation types from R, with the last node in NodeQ and add them to relationalGraph. A number of paths are created between each concept in the relationalGraph and the current path. The new created paths are saved in PathQ. If all paths in PathQ are checked and Kpaths does not exist, the function returns ‘Not found’. Let us consider Sword= ‘Egypt and Eword =Israel, and $k < 4$. Figure 6 illustrates the process of discovering the knowledge-path between the two concepts ‘Egypt’ and ‘Israel’.

Function discover knowledge-paths

```
def discover-knowledge-path(relationalGraph, Sword,
Eword,k,R):
    PathQ = []
    Kpaths=[]
# push the first path into the PathQ
    PathQ.append([Sword])
    for sen in Sword.sense():
      while PathQ:
# get the first path from the PathQ
        NodeQ = PathQ.pop(0)
# get the last node from the NodeQ
        node = NodeQ[-1]
# path found
        if node == Eword:
            Kpaths.append(NodeQ)
            return Kpaths
        else:
          if len(NodeQ) < k:
              s=list()
              for key, value in R.iteritems():
                  n=value
# get all concepts have relations from R with node and
have the same sense of node
                  x=n(node, sen, key)
                  s=s+x
                  relationalGraph[node]=s
              end for
          end if
        end if
# enumerate all adjacent nodes, construct a new path and
push it into the queue
        for adjacent in relationalGraph.get(node,[ ]):
          new_path = list(NodeQ)
          new_path.append(adjacent)
          if len(new_path) < k:
            PathQ.append(new_path)
          else:
            break
          end if
        end for
      end while
    end for
    if !(Kpaths)
      return 'Not found'
```

Figure 5. Discovering Knowledge-paths function.

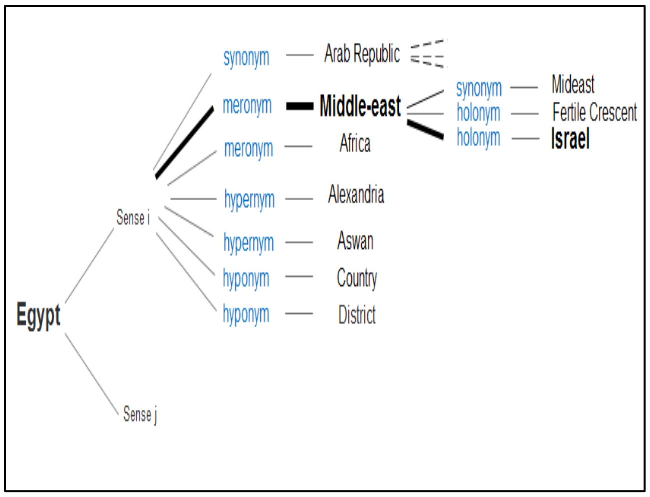

Figure 6. Knowledge Path example.

The following path is the knowledge path returned by the algorithm. **Egypt (***meronym)* **Middle-east** (*holonym)* **Israel.**

IV. EXPERIMENTAL EVALUATION

This section describes an evaluation of our methodology based on the obtained results. We chose an interesting article from the media as the target text. Its title is "*Egypt: Ex-ruler Hosni Mubarak, accused in deaths of hundreds, cleared of charges*" [12]. The article was released on November 30, 2014 and has 45 sentences and 152 concepts where the concepts are only nouns. The used ontology engine is WordNet, version 1.7. We select the first 42 concepts from 152 concepts as the most important text concepts n`. The used reference text is Wikipedia. The reference text was about 'Hosni Mubarak' [13]. It was selected based on the first concept in n_c. Table II shows the number of nodes in the three graphs Target-Text Graph (TG), Semantic-Graph (SG), and illuminated-Semantic-Graph (ISG).

TABLE II NUMBER OF NODES IN THE THREE GRAPHS

	TG	SG	ISG
T	42	42	42
OE	0	7	13
R_T	0	0	6
Total	42	49	61

Figure 7 shows the relations types of the three graphs with the ontology engine (OE) relations (synonym, hyponym and meronym). We see that the ISG set has the largest number of relations followed by SG set, while the TG has no relations. SG set shows that there are zero synonym, 18 hyponym, and 2 meronym relations which mean that using OE helps in discovering some of the hidden relations among T concepts. In ISG, three new synonym relations appear among T concepts, the number of hyponym relations reached to 29, while the number of meronym jumped to 4. Therefore, using OE and RT concepts discover more hidden relations among T concepts.

Humans discovered that not all the gained knowledge paths are correct and some of them are considered noise paths. Figure 8 shows a breakdown of the obtained geometric-paths in the three graphs. Evidently, the number of the correct knowledge paths in SG and ISG has risen gradually, while the number of noise paths is stable. These results confirm that our system is suitable for discovering the knowledge paths among *T* concepts.

Figure 9 shows an overview of the resulting of five well-known graph parameters (disconnected-components, giant component, average non-giant component, average degree, and diameter) on the three graphs. In TG, the 42 concepts are disconnected-components because there is no knowledge paths among them thus the average size of non-giant component is 1, while all the rest of graph parameters are zero. In SG, the number of disconnected-components is sharply decreased to 23. This refers to that the use of OE helps in discovering some relations among some of the *T* concepts by adding knowledge paths among them. As seen, the size of the giant component jumped to 7 and the average size of non-giant component [14] is 1.5 which means that 14.3% of the concepts in this graph including T and OE concepts are reachable from one another. The average degree [14] increased to 0.82 that means there is a high number of relations appear among the *T* concepts, while the diameter is 6, that refers to the longest knowledge path between any two *T* concepts. For ISG, all the results of the graph parameters are improved as the following: The number of disconnected-components is decreased to 18, which means that adding the OE and the R_T concepts adds more relations and knowledge among the *T* concepts. The size of giant component including T, OE, and R_T concepts increased to 12, which is almost 19.7% of the concepts in this graph and the average size of non-giant component is 1.9 that refers to the more reachability among the concepts. Thus, more of the hidden relations among the *T* concepts appeared and more knowledge added. Average degree increased to 1.11, which points to the much high connectivity among the T concepts, whereas the diameter reached to 9.

The SG and the ISG of our experiment are shown in Figure 10 and Figure 11. In Figure 10 the white nodes show the T concepts, the dark gray nodes represent the OE concepts, and edges among them represent the paths. The solid black edges illustrate the correct knowledge paths and the dashed edges depict the noise paths. Additionally in Figure 11, the light gray nodes are the new concepts added from RT and the solid gray edges show the knowledge paths among T and RT concepts. The double circle nodes are the shared nodes between T and RT. In Figure 10 and Figure 11, H refers to the Hyponym relation, M refers to the Meronym relation, and S represents the Synonym relation.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a computerized human text comprehension technique by automatic reading reference texts, that gives a deep comprehension about a specific text and demonstrate how it can improve the text comprehension. In the process, we used the SVD algorithm to extract the most important concepts in the target text and the reference text to generate the illuminated-Semantic-Graph using ontology engine. The results show that the system improves the text comprehension by adding relations and new knowledge among the concepts. Although the system succeeded in finding the knowledge paths, it has a limitation with filtering the correct knowledge paths from the noise paths. Alternatively, we perform the filtration manually. We intend to handle this issue automatically in the future work.

REFRENCES

[1] Y. Kang, L. Zhou, and D. Zhang “An integrated method for hierarchy construction of domain-specific terms” IEEE/ACIS 13th International Conference on. IEEE, 2014, pp.485-490.

[2] P. Cimiano, and S. Staab "Learning concept hierarchies from text with a guided hierarchical clustering algorithm”, ICML 2005 workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods, Bonn, Germany. 2005, pp. 6-16.

[3] G. Hirst, and D. St-Onge, “Lexical chains as representations of context for the detection and correction of malapropisms” In Fellbaum, C., ed., WordNet: An electronic lexical database, 1998, pp. 305–332.

[4] R. Girju, A. Badulescu, and D. Moldovan, “Automatic Discovery of Part- Whole Relations. Computational Linguistics”, Vol. 32, 2006, pp.83-135.

[5] C. S. Yadav, A. Sharan, and M. L. Joshi, "Semantic graph based approach for text mining." Issues and Challenges in Intelligent Computing Techniques (ICICT), International Conference on. IEEE, 2014, pp. 596-601.

[6] D. Tu, L. Chen, and G. Chen. "Automatic multi-way domain concept hierarchy construction from customer reviews." Neurocomputing 147, 2015, pp.472-484.

[7] D. Tu, L. Chen, and G. Chen, “WordNet based multi-way concept hierarchy construction from text corpus”, Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013, pp. 1647-1648.

[8] M. A. H. Taieb, M. Ben Aouicha, M. Tamer, and A. Ben Hamadou "Wikipedia category graph and new intrinsic information content metric for word semantic relatedness measuring" Data and Knowledge Engineering. Springer Berlin Heidelberg, 2012, pp. 128-140.

[9] M. Lipczak, A. Koushkestani, and E. Milios "Tulip: lightweight entity recognition and disambiguation using wikipedia-based topic centroids" Proceedings of the first international workshop on Entity recognition & disambiguation. ACM, , 2014, pp. 31-36.

[10] M. S. Han, and E. E. Mon. "Computing Semantic Relatedness using Wikipedia Taxonomy by Spreading Activation" International Conference on Advances in Engineering and Technology (ICAET'), Singapore, 2014, pp. 49-53.

[11] G. H. Golub, and C. Reinsch. "Singular value decomposition and least squares solutions" Numerische Mathematik 14.5, 1970, pp. 403-420.

[12] J. Hanna, S. Sirgany and H. Yan. (2015, July 25). Egypt: Ex-ruler Hosni Mubarak, accused in deaths of hundreds, cleared of charges. CNN. Reterived from: http://www.cnn.com/2014/11/29/world/meast/egypt-mubarak-trial.

[13] Hosni Mubarak (2015). From Wikipedia. Retrived July 25, 2015, from: http://en.wikipedia.org/wiki/Hosni_Mubarak.

[14] M. S. Hardas “Segmentation and Integration in Text Comprehension: A Model of Concept Network Growth” Diss. Kent State University, 2012.

[15] L. Wilkinson, A. Anand, and R. Grossman. "Graph-Theoretic Scagnostics", INFOVIS. Vol. 5, 2005, pp.21.

[16] A. Babour and J. I. Khan. "Tweet Sentiment Analytics with Context Sensitive Tone-Word Lexicon" Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Vol. 01. IEEE Computer Society, 2014, pp. 392-399.

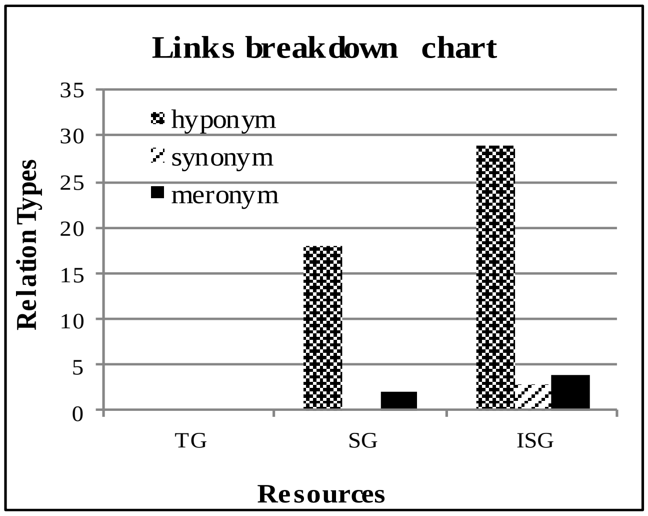

Figure 7. OE relation analysis for the three graphs.

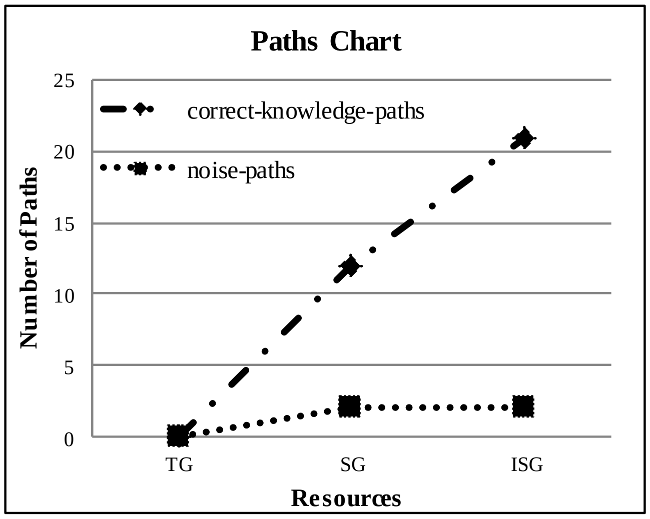

Figure 8. Paths analysis for the three graphs.

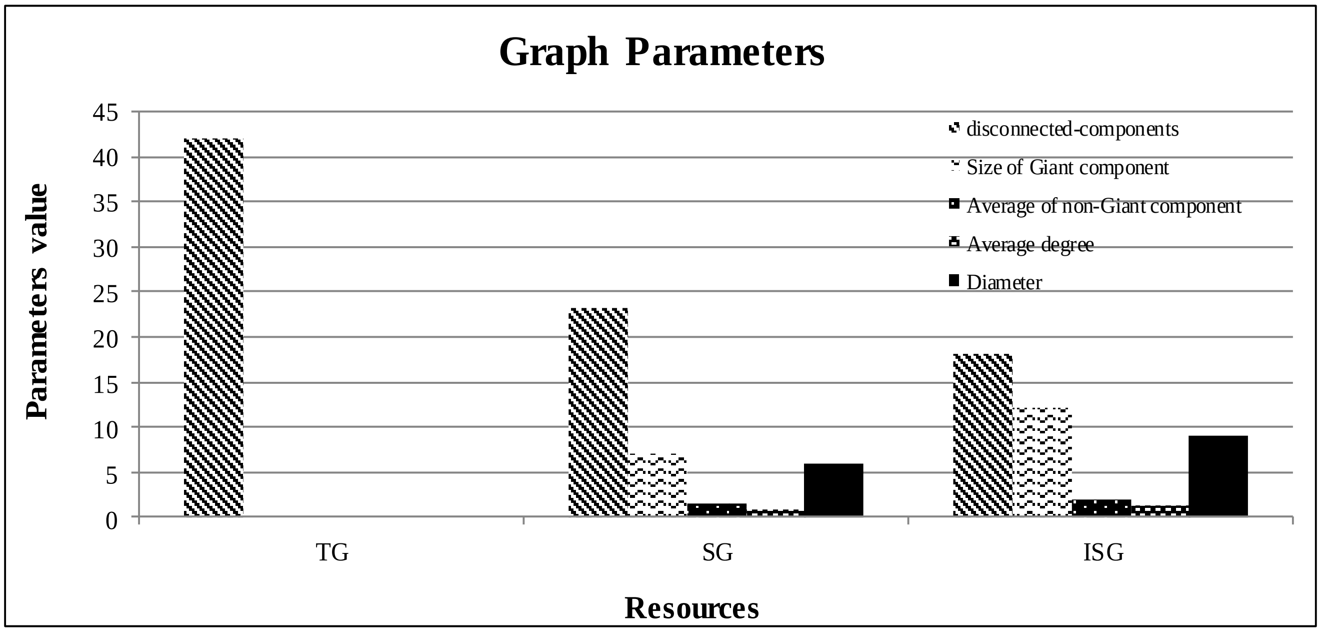

Figure 9. Measuring graph parameters for the three graphs.

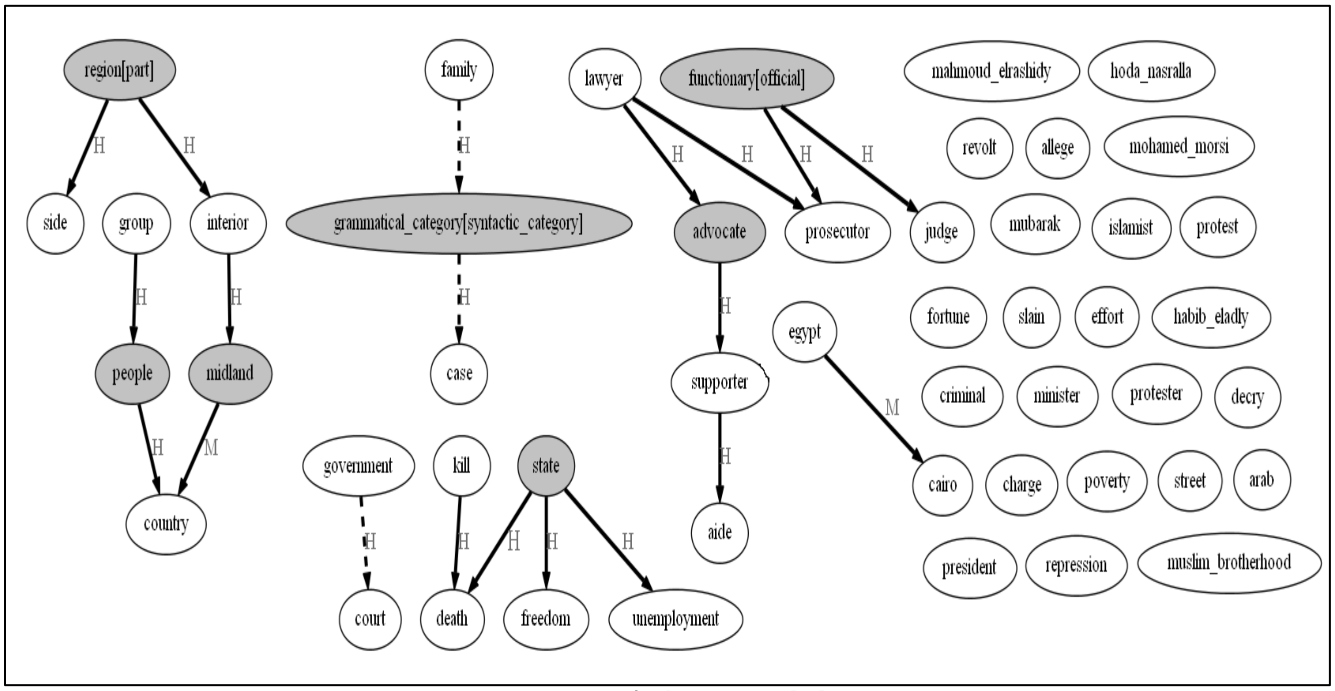

Figure 10. SG for the Target text [12].

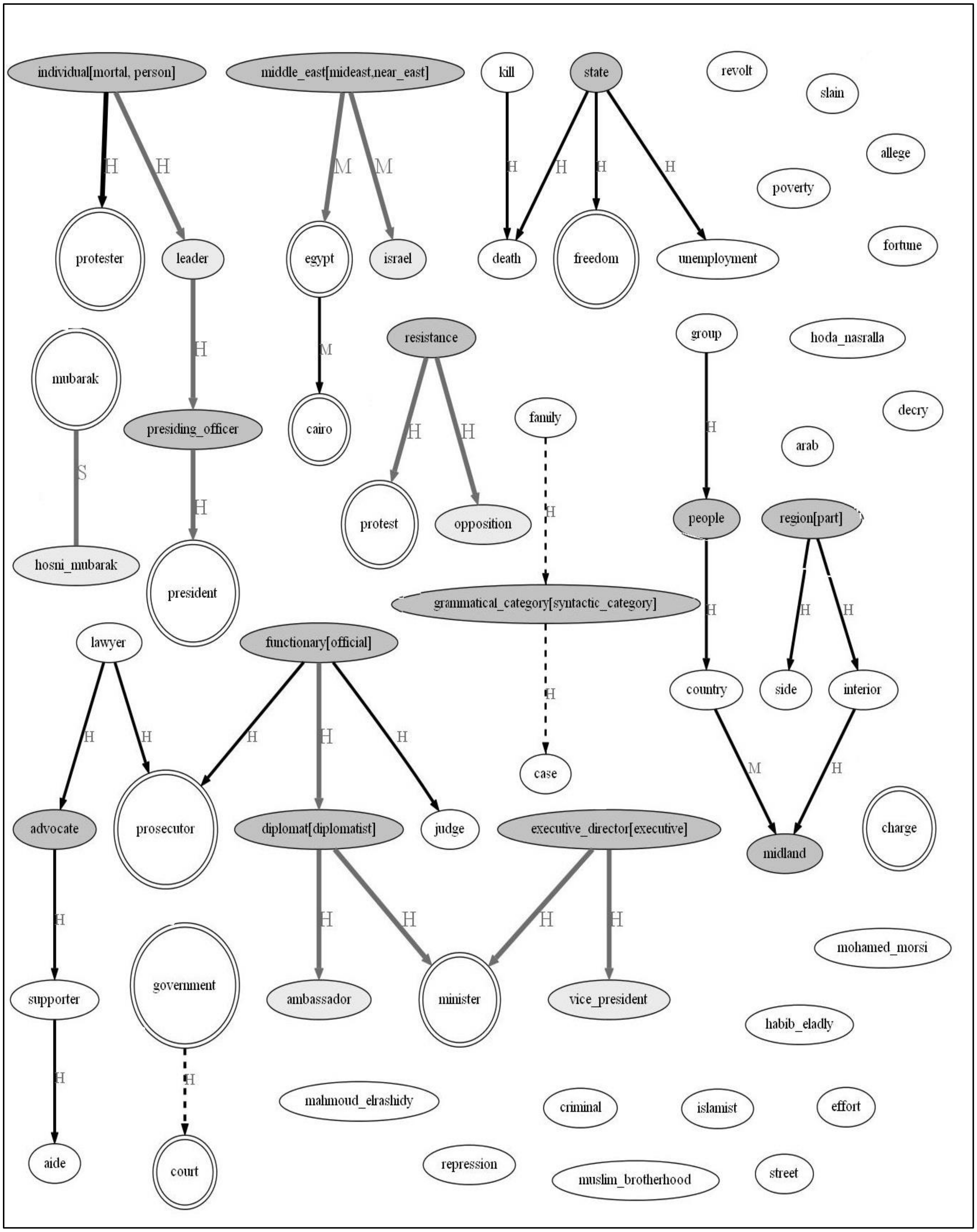

Figure 11. ISG for the Target text [12].