

English2MindMap: an Automated System for MindMap Generation from English Text

Mohamed Elhoseiny and Ahmed Elgammal
Computer Science Department
Rutgers, The state university of New Jersey
Piscataway, NJ, USA
{m.elhoseiny,elgammal}@cs.rutgers.edu

Abstract—MindMapping [1] is a well-known technique used in note taking and is known to encourage learning and studying. Besides, MindMapping can be a very good way to present knowledge and concepts in a visual form. Unfortunately there is no reliable automated tool that can generate MindMaps from Natural Language text. This paper fills in this gap by developing the first evaluated automated system that takes a text input and generates a MindMap visualization out of it. The system also could visualize large text documents in multilevel MindMaps in which a high level MindMap node could be expanded into child MindMaps. The proposed approach involves understanding of the input text converting it into intermediate Detailed Meaning Representation (DMR). The DMR is then visualized with two proposed approaches; Single level or Multiple levels which is convenient for larger text. The generated MindMaps from both approaches were evaluated based on Human Subject experiments performed on Amazon Mechanical Turk with various parameter settings.

Keywords-MindMapping; Natural Language Processing; Text Mining; Text Visualization

I. INTRODUCTION

MindMapping is a technique that was developed by Tony Buzan in the 1960s [1]. It is a powerful pictorial technique for representing knowledge, concepts and ideas. Figure 1 represent linear text of Shakespeare's life, and its corresponding manually drawn MindMap respectively.

With today's vast and rapid increase in the use of electronic document readers, smart phone, tablets, there is an eminent need to develop tools for visualizing and summarizing textual contents. MindMapping is not only a note-taking tool, but also a very powerful tool for text summarization and visualization. Converting a text paragraph to a MindMap would provide an easier way to visually access the knowledge and ideas in the text.

One of the reasons why many people do not use MindMaps is that making MindMaps needs huge extra mental efforts and full concentration for a very long time specially if someone wants to make MindMaps for a long script or a book. Besides, only few people are creative enough to draw good MindMaps.

There is no practical approach available to automatically convert text documents to a MindMap representation. therefore, There is no industrial tools released to do this task. This

William Shakespeare (1564-1616)

Very little is officially known about Shakespeare; but scholars have pieced together a reasonably comprehensive picture of his life from his marriage to Anne Hathaway in 1582 to his Christenings of his 3 children. And most reputable critics ascribe to him the authorship of the major portion of thirty eight of the world's most respected plays several excellent poems, and some 154 sonnets.

William achieved success largely on his own. He apparently never attended college. Successive purchases and sales of agricultural products and parcels of land near Stratford must have provided Shakespeare with greatly increased capital, which, when reinvested paid him steady income for many years. This gave him the freedom and time to concentrate on his first loves: acting and writing.

In 1594-1595 William performed before Queen Elizabeth and his name became widely recognised. Shakespeare grew in public stature when he became one of the owners of London's Globe Theatre in 1599.

For Study, Shakespeare's works can be divided into six separate, somewhat chronological sections; Early works, Major Histories, The problem plays, Tragedies, The Roman Plays, and The Late Romances. His comedies seem to be interspersed throughout these divisions.

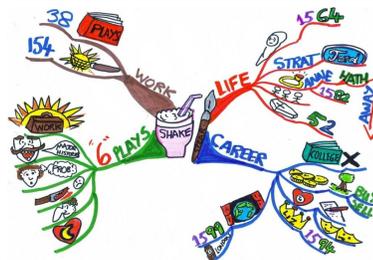


Figure 1: Shakespeare life MindMap (text description on the top while the hand drawn MindMap on the bottom) [2].

is because the current approaches are inapplicable for large text visualization. Hence, the goal of this work is to fill in this gap by developing the first evaluated system that takes a text input and generate a MindMap visualization out of it. The tool is constructed to visualize large text documents in multilevel MindMap in which a high level mid map node could be expanded into a child MindMap.

The contributions of this paper are: 1) Novel English2MMap system Architecture. 2) Novel Multilevel approach for generating the MindMap. 3) Two approaches for visualization which by retrieving relevant pictures from the web (two approaches are presented and evaluated). 4) First comprehensive evaluation of automated MindMap system by human subjects.

The rest of this paper is organized as follows. Section II presents the related literature. Sections III to VI describe

the proposed approach and techniques used to build up the system. Section VII describes the evaluation procedures and shows the results using different system parameters. Section VIII presents our conclusion and the future work.

II. RELATED WORK

Our formulation is a hybrid between text summarization and visualization. In the context of text summarization, the top level MindMap serves as a summarized version of the document. Meanwhile, it is a visualization technique, where each level is presented by the corresponding MindMap. Text summarization has been investigated over the past five decades [3]. The remaining of this section presents the related work in text visualization and MindMapping tools.

A. Text Visualization

There are few relevant approaches in text Visualization. Some of the state-of-the-art text visualization techniques build on tag clouds. Examples include work conducted by IBM in Wordle, which is a web-based text visualization tool that creates a tag-cloud-like displays with careful attention to typography, color and decomposition [4]. The generated clouds give greater prominence to words that appear more frequently in the source text. Van Ham et al. [5] presented a technique called PhraseNet, which diagrams the relationships between different words used in a text. It uses a simple form of pattern matching to provide multiple views of the concepts contained in a book, speech, or poem.

Wordle and PhraseNet might look very similar to our problem, however, three significant differences differentiate our work. First, their algorithms do not involve deep analysis of the text, so the building block is a single word. Second, each node is visualized with the word itself, nevertheless in MindMaps, visual nodes are mapped into pictures of a concept. Third, our system can generate multilevel representation, which is not supported by either Wordle or PhraseNet.

A more relevant applications is the automatic conversion of text to 3D animation. Ma [6] proposed Lexical Visual Semantic Representation (LVSR), which connects linguistic semantics to the visual semantics and is suitable for action execution. This application involves deep analysis of the text converting it into user defined concept by which it converts the text into actions to be performed by 3D models.

B. MindMapping Tools

There exist several tools that can help draw a MindMap as just an editing canvas (e.g. [7]). However, the most relevant work was conducted by Hamdy et al. [8]. They presented a prototype for the MindMap Automation, however, it was tested with examples of few sentences and then it was applied in a mobile application in [9]. There are two critical drawbacks in this approach. (1) It is almost not evaluated. (2) It supports only single level MindMaps. These limitations made the approach incapable of representing information in larger text.

III. ENGLISH2MINDMAP SYSTEM ARCHITECTURE

Figure 2 illustrates the architecture of English2MindMap system, where the blocks are color-coded according to the contribution in each block. Gray blocks almost used existing approaches and/or technologies (e.g., the TPM), while green blocks constitute the main contribution. The following subsections describe the functionality of each component. The contributions are then detailed in the following sections.

A. Text Preprocessing Module (TPM)

This is the first phase in the system, it extracts sentence-wise information from raw text. It takes as an input the English plain text, and then generates for each sentence a parse tree, discourse analysis results, and the intended sense for each word in the sentence. Internally, this block is divided into five components: morphological analysis [10]; parsing [11]; syntactic structural analysis [12], which reduces possible parse trees of a sentence into a single tree; discourse analysis [13], which includes anaphora resolution; and finally Word Sense Disambiguation (WSD) [14] determines the intended sense for each word in the sentence based on WordNet [15]. In this stage, we utilized the current approaches for such a well-studied phase in NLP.

B. Detailed Meaning Representation (DMR) Generation

This component is responsible for representing the text in a meaningful form. DMR consists of a complex structure filled according to the ontology in the form of feature-value pairs. This component takes the output of TPM, from which it generates a semantic graph of frames [16]. The challenge is to build an optimal representation that has a strong visual cohesion and harmony.

C. Multi Level MR Generation (MLMR)

This phase generates a multi-level meaning representation, where it takes the DMR as an input and produces higher-level meaning representation recursively and depending on the concepts stored in the ontology. This component involves node ranking which measures node persistence in the high levels of the meaning representation. It also performs semantic grouping based on both the relevance of nodes in MR, measurement of semantic distance measure between nodes, and automatic determination of the number of levels.

D. MindMap Generation

The core function of this component is to convert either the DMR or MLMR into its final visual form by replacing conceptual information by a visual content (i.e.image). It then dynamically allocates the generated MindMap on the computer screen. We investigated several ways to represent visual concepts by images. This includes size variations (e.g. Small, Medium, All) and type variations (e.g. All, ClipArt, LineArt).

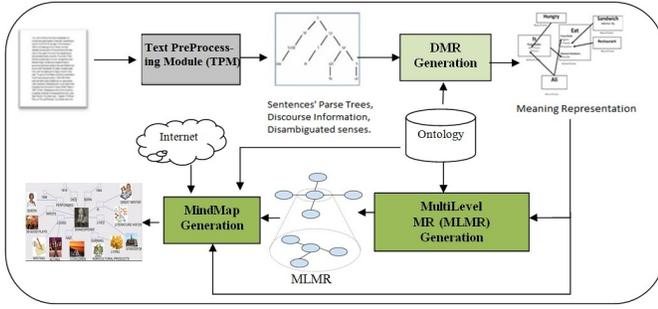


Figure 2: English2MindMap System Architecture

IV. DMR GENERATION

Designing a meaning representation for NLP involves determining its content and its representation. The structure of the representation is shown to be based on the needs for composing the primitives in different ways. The following subsections illustrate the meaning representation, used in the system, and how it is generated.

A. Meaning Representation (MR)

In our system, the meaning representation consists of conceptual frames and relations between these frames. Hence, two types of frames are included in our system (Entity frames and Action Frames). Entity frames represent entities mentioned in the text (persons, objects etc) , while Action Frame involves actions in the text. Possible relations between the entities are presented in Figure 3. Hence, the final output of the DMR generation is a graph whose nodes are the frames (Action Frames or Entity frames) while edges are the relations.

B. MR Generation

As an output of the TPM, for each sentence the following is available (a) The parse tree (b) Discourse information (c) The disambiguated word sense for each word. To generate the DMR, the parse tree of each sentence is traversed and the grammatical structure is followed to defined subjects, objects, and verbs. They are then mapped into Entity or Action Frames (avoiding redundancy using discourse information) and relations. The case roles of the frames and relations used in the system are defined in Figure 3.

Having generated frames (Entity frame or Action frame), each of them is mapped to the corresponding ontological concept (directly retrieved from the ontology as illustrated in Figure 2). Hence, conceptual attributes are filled up whenever appropriate.

V. MLMR GENERATION

The main goal of this phase is to group each set of related actions or entities in the MR into a common concept, so that it is easier for the reader to identify and understand. Without this phase, if an input is of a quite significant size

CaseRole Enum	wCr	RCr	DomainRelation... Enum	wDr	RDr	TemporalRelati... Enum	wTr	RTr
Agent	100	2.0	ExpectedResult	50	0.8	After	30	0.3
Theme	90	1.5	UnexpectedResult	50	0.8	Before	30	0.3
Cotheme	80	1.0	Reason	60	0.8	Concurrent	30	0.4
Accompanier	70	0.8	Contrast	60	0.8			
Instrument	60	0.6	Comparison	60	0.8			
Source	50	0.8	Condition	60	0.8			
Destination	50	0.8	PossibleResult	60	0.8			
Path	100	0.5	Concessive	60	0.8			
location	20	0.8	Enablement	60	0.8			
time	20	0.8	manner	50	0.6			
Means	50	0.5	place	20	0.6			
OwnerOf	50	0.5	InclusiveOR	60	0.8			
Beneficiary	90	1.5	ExclusiveOR	60	0.8			
purpose	60	0.6	Addition	60	0.8			
Action	60	0.6	Enumeration	60	0.8			

Figure 3: Relations in Meaning Representation and their weights

involving many concepts, the output would be quite unclear and unorganized.

MLMR generation performs semantic grouping on the DMR and build up relationships between the groups such that a MindMap can be built up in Parent-children relations. In other words, each MindMap node can be expanded to child MindMaps.

MRSA (Meaning Representation Summarization Algorithm), detailed later in this section, is the basic building block to generate MLMR from the DMR and the ontology. The process MRSA summarizes the meaning representation based on the ontology and the importance of the frames. Furthermore, each frame of the summarized MR is connected to regions (i.e., Frames and Relations) in the DMR that are related to that frame. Frames in the summarized MR that map to more than one frame in the input MR are marked as *group frames*. MLMR generation, based on MRSA, is accomplished by iteratively finding sub MRs to summarize as presented in Algorithm 1.

Algorithm 1 MLMR Generation(DMR)

```

topMR ← MRSA(DMR)
Display MindMap of topMR to the user and Highlight
group frames
while true do
    Wait until user select group frame, exit if user choses
    to exit
    currentMR ← MRSA(regionInDMR(Selected Frame))
    Display MindMap of currentMR to the user and High-
    light group Frames
end while

```

MLMR could be generated without user interaction by iteratively expanding group frames that didn't appear in previous expansions. Clearly, MLMR depends mainly on MRSA, which could be summarized as follows. The intuition behind MRSA is to measure the significance of frames by a weight assignment phase. It then selects the most important frames (we call them main frames) by clustering, based on the assigned weights. Afterwards, the main frames persist at the top level, with surrounding actions conceptually

grouped according to a conceptual metric (i.e., distance between the concepts inside the ontology hierarchy). This step is achieved by conceptual based partitioning phase. The following subsections present the details of MRSA phases.

A. Weight Assignment

The goal of this step is to determine the importance of each frame, in the MR, based on their associations with other frames in the DMR. The weight assigned to each frame indicates its level of significance; the higher the weight, the more significant the frame is. Each of the case roles, domain relations, and temporal relations (shown in Figure 3) is assigned a constant weight depending on its type (for example Agent may have score different from Location or Reason). The weight per Entity Frame is the sum of the values of its surrounding relations (Equation 1).

$$WEF_k = \sum_i N_{CR_i}^k w_{CR_i} + \sum_i N_{DR_i}^k w_{DR_i} + \sum_i N_{TR_i}^k w_{TR_i}, \quad (1)$$

where WEF_k is the weight of Entity Frame k and $N_{CR_i}^k$ is the number of Case Role relations of type i in the frame k . Similarly, $N_{DR_i}^k$ and $N_{TR_i}^k$ are the number of Domain Relations, Temporal Relations of type i in the frame k respectively. w_{CR_i} , w_{DR_i} and w_{TR_i} are the constant weights assigned to CaseRole, Domain and Time Relations of type i (Figure 3). The weight of Action frames is then deduced from the weights of its neighbor frames and relations (Equation 2).

$$WAF_k = \sum_i (R_{CR_i} \sum_j w(FCR_{ij}^k)) + \sum_i (R_{DR_i} \sum_j w(FDR_{ij}^k)) + \sum_i (R_{TR_i} \sum_j w(FTR_{ij}^k)), \quad (2)$$

where WAF_k is the weight of Action Frame k , R_{CR_i} is a float $\in (0, 2]$ that defines the weight of the case role of type i (Figure 3). $w(FCR_{ij}^k)$ is the weight of the j^{th} frame connected to frame k with caserole of type i . Similarly, R_{TR_i} and R_{DR_i} are the corresponding ratios for temporal and domain relations while $w(FDR_{ij}^k)$ and $w(FTR_{ij}^k)$ are the corresponding weights of frames related to frame k with temporal and domain relations respectively.

B. Weight-based Partitioning

The goal of this step is to identify the main entity frames in the MR. The action frame weights, obtained from the weight assignment, are partitioned into clusters using K-Means++ [17]. We used the method in [18] to select the best K.

C. Concept-based Partitioning

The goal is to group related information under one common concept according to an ontology. The associated actions of each main entity frame (extracted from the top cluster of the weight-based partitioning) are passed through the concept-based partitioning and a list of concepts with

their corresponding frames is returned. If an action frame of a main entity frame was not grouped with any of the other action frames, the associated entity frames of that action frame are grouped using the concept-based partitioning. So, the new text meaning representation will contain (1) the main entity frames, (2) the grouped concepts associated with them as new action frames, (3) the ungrouped action frames around the main entity frames and (4) the grouped concepts of noun frames around them as new noun frames. The following steps illustrates how to conceptually partition concepts around given a main entity frame.

- 1) Group frames of the exact concept (i.e. ontological distance =0).
- 2) If the count of the groups in 1 is $> G_{th}$ (We used $G_{th}=3$), perform agglomerative hierarchical clustering (AHC) until count of the groups is $\leq G_{th}$.

A significant advantage of this model is that each high level frame could be expanded into child files through (AHC) and the algorithm is to investigate and analyze the selected frame and its neighbors and try to do conceptual partitioning if convenient as illustrated in Figure 4.

VI. MINDMAP GENERATION

This phase is concerned with the conversion of either the DMR or MLMR to a MindMap that contains images for visual frames. Visual frames are detected by checking the ontology if the given frame's concept is visual. After the frame is verified as visual from the ontology, query text is generated for the visual frame to retrieve a relevant image from Google image search. Figure 4 shows single level and multilevel outputs of our system.

A. Query Generation and Image Web Retrieval

We have implemented two ways to generate the query. In the first approach, the given frame is associated with its filled attributes. For instance, if there exist an entity frame (Ball: Color= Red, Size=small), the generated query will be "small red ball". On the other hand, The second approach involves frame combination. An example to illustrate this point, if we have a sentence like "Shakespeare performed before the queen in December". In this case if the web image search on "the queen", an irrelevant image will appear with high probability. However, if the query combines "Shakespeare queen". The query will return a picture of "Elizabeth queen" which is meant by in this sentence. That's why we tried to combine frames in the query. We implemented concept combination as follows. Given frame adds to the generated query generated by the first approach, the name of frame within depth 2 (i.e., relevant frame) if and only if it's significantly more important. Relative importance of $frame_i$ with respect to $frame_j$ is mathematically defined by $WF_i/WF_j > th$ where WF_i is the weight of frame i , WF_j is weight of frame j , th is a threshold parameter ($th =6$ in our experiments).

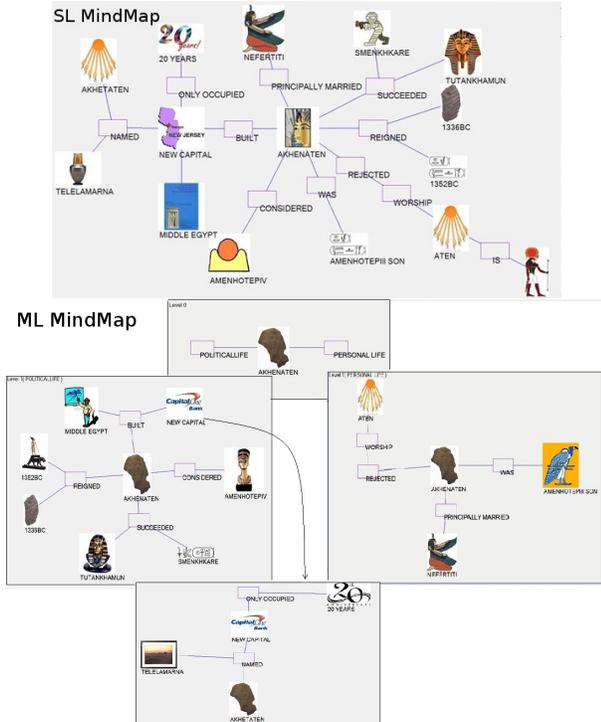


Figure 4: Akhentan Single Level and Multi Level MindMaps

Finally, The images displayed in the MindMap are obtained using Google Image Search (GoIS). We used the GoIS API to retrieve the relevant images using the generated query and the first image in the results is chosen.

B. Automatic Layout allocation

In this system, we achieved an acceptable layout by adopting spring model [19]. We run the spring model 10 times with different initial seeds selected randomly. then, we select the layout with the minimum value of the cost function in [19].

VII. EXPERIMENTAL RESULTS

This section presents the evaluation methodology and experiments of our work based on 4900 Mechanical Turk rating responses with different system parameters.

A. Historical figures dataset

Since this kind of systems has not been comprehensively evaluated, there is no existing dataset for MindMap Automation. Hence, We created a dataset to evaluate the system. We chose Historical Figures articles (e.g., Shakespeare, George Washington, etc) to build our dataset. 35 historical figures were chosen from the BBC historical figures section [20] of size between 150 words to 250 words. We used the system to generate 455 different MindMaps from the 35 chosen articles. The 455 cases were evaluated based on Human Subject Ranking of 4900 MTurk workers' responses using the following 5 question survey as follows. (1) To what extent the generated output represent the text (Regardless of

the pictures) ? Grade 1-5. (2) To what extent the generated pictures are relevant? Grade 1-5. (3) How many missing Actions in the shown diagram (if no missing Actions, please put 0)?. (4) How many missing Entities in the shown diagram (if no missing Entities, please put 0)?. (5) How many repeated Entities/Actions in the shown diagram (if no missing Entities/Actions, please put 0)?

The evaluation is partitioned into three experiments to evaluate the effect of changing the system parameters. The system was evaluated upon three main variations. (a) GoIS Parameters (Size, Image Type) for single level MindMap in Experiment 1 (b) Concept combination in Experiment 2 (c) Multilevel MindMaps in Experiment 3. Regarding the ontology used in this evaluation, we have created an ontology of 1150 concepts that spans the senses in the documents in the dataset. The ontology contains mainly concepts related to Work, Personal Life, and Political Life, which are suitable for the articles.

The responses of the experiments were evaluated based on three metrics: (1) Mean, (2) Standard Deviation to indicate the stability of the response, and (3) For grading questions (i.e., Q1,2), Satisfaction Ratio (SR) that is defined as $SR = \frac{\#ofresponses \geq 4}{\#ofresponses}$.

B. Experiment 1 (315 cases, 3150 responses)

Experiment 1 aims at evaluating user satisfaction with Single Level MindMaps and to test whether their responses are biased by varying the displayed image parameters. We tested nine variations: three different image types (All [any image], ClipArt, LineArt) and three sizes (All, Auto, Small). The auto size mode is a functionality we have implemented to determine the image size according to the weight of the frame to be visualized. Our implementation involves medium size image, biggest suitable size for MindMaps, if the number of relations to it is greater than 6, otherwise, a small image is assigned to the frame. The MindMap generation phase in this experiment did not involve concept combination (which is evaluated in Experiment 2). Overall, we have 315 (9 × 35) cases with 10 MTurk responses each (3150 responses). Table I (top part) presents the user response statistics aggregated over all of the 9 variations. The results indicate general satisfaction by the MTurk workers.

Q1 and 2 are the most relevant questions to the appearance of the generated MindMap and the generated images. The results of these questions are grouped for each of the 9 variations in this experiment. The grouped results are almost the same as Experiment 1 results for each type. This indicates that changing the image type or size almost do not affect the responses of the MTurk workers. In other words, this responses were not biased to specific image type of size.

C. Experiment 2 (105 cases,1050 responses)

Experiment 2 focuses on the user satisfaction of the MindMaps in case of Concept Combination mode (described

Experiment 1					
	Q1	Q2	Q3	Q4	Q5
Mean	4.42	4.34	0.54	0.59	0.48
StdDev	0.84	0.87	1.58	1.74	1.68
SR	0.86	0.84			
Experiment 2					
	Q1	Q2	Q3	Q4	Q5
Mean	4.51	4.27	0.45	0.5	0.23
StdDev	0.65	0.82	1.25	1.47	0.58
SR	0.92	0.79			
Experiment 3					
	Q1	Q2	Q3	Q4	Q5
Mean	4.48	4.34	0.59	0.61	0.25
StdDev	0.87	0.94	1.51	1.55	0.93
SR	0.89	0.84			

Table I: Human Subject Responses for Experiments 1, 2, 3

in subsection VI-A). To test if the MTurk workers were biased to the pictures, we have created Concept Combination cases with 3 variations of image types. This results in 105 cases (3×35 historical figures). Each case was evaluated by 10 different MTurk anonymous workers (1050 responses).

Table I (middle part) shows summarized results for the experiment. Compared to Experiment 1, there is an improvement of 2% in the mean and 7% in the satisfaction ratio for Q1 with the use of concept combination. However the satisfaction ratio of Q2 (pictures satisfaction) decreased by 10%. This indicates that the retrieved image using direct query is relatively better than retrieved image using Concept Combination query. The results were also grouped by the 3 variations of Experiment 2, which indicate similar behavior to Experiment 1 (i.e. No bias).

D. Experiment 3 (35 cases, 700 responses)

The purpose of this experiment is mainly to rate the partitioning of the information in the multilevel MindMaps and not the evaluation of the retrieved images (evaluated in Experiment 1 and 2). Hence, we selected one case for each historical figure (with the best images retrieved in Experiment 1). Q2 was changed to evaluate of conceptual grouping instead of the pictures as they were already evaluated in Experiments 1, 2. Question 2 became "To what extent are you satisfied with the structure of the MindMap (Clarification, the classes like Work, Life, Political Life and information included in the correct class. As a wrong instance, information like the historical figure's birth or death is classified under Work)? Grade 1-5". To evaluate it interactively, we have generated 35 interactive flash file (one for each historical figure). Each flash file was evaluated by 20 MTurk Users (i.e., 700 responses).

Results are shown in Table I (bottom part). There is 1.4% and 3% improvement in the mean and SR respectively compared to the single-level MindMaps. The results show the satisfaction of the users with the hierarchical representation of the MindMap. While both single-level and multilevel approaches give satisfactory outputs, the multilevel approach

is the only way to represent all information in large text, where single level generation is not applicable.

VIII. CONCLUSION AND FUTURE WORK

We have designed and implemented an automated tool that takes English text as input and generates a Mind Map visualization out of it. The system was comprehensively tested under different parameter settings by MTurk Human Subjects and high satisfaction rates have been recorded. Hence, we aim to extend the system such that it's reliable in handling very large text (e.g., a book) and also to try different approaches of Concept Combination. We will also work on enhancing the performance of the system to handle large text in reasonable time.

REFERENCES

- [1] B. B. Tony Buzan and J. Harrison, *The Mind Map Book*. BBC Archive, 2010.
- [2] Mind-mapping.co, *Shakespear Mindmap*, 2012 (accessed February, 2012). [Online]. Available: <http://www.mind-mapping.co.uk/mind-maps-examples.htm>
- [3] D. Das and A. F. T. Martins, "A survey on automatic text summarization," *Literature Survey for the Language and Statistics II Course at CMU*, 2007.
- [4] F. Viegas, M. Wattenberg, and J. Feinberg, "Participatory visualization with wordle," *TVCG*, 2009.
- [5] F. van Ham, M. Wattenberg, and F. B. Viegas, "Mapping text with phrase nets," *TVCG*, 2009.
- [6] M. Ma, "Automatic conversion of natural language to 3d animation," Ph.D. dissertation, University of Ulster, 2006.
- [7] novamind.com, *NovaMind tool*, 2012 (accessed February, 2012). [Online]. Available: <http://www.novamind.com/>
- [8] A. Hamdy, M. H. ElHoseiny, R. E. Sahn, S. Samier, and E. Kamal, "Mind maps automation (mma) system," in *SWWS*, 2009.
- [9] R. Kudelic, M. Konecki, and M. Malekovic, "Mind map generator software model with text mining algorithm," in *ICITI*, 2011.
- [10] Z. Youzhi, "Research and implementation of part-of-speech tagging based on hidden markov model," in *Asia-Pacific Conference on Computational Intelligence and Industrial Applications*, 2009.
- [11] M.-C. de Marneffe, B. MacCartney, and C. D. Manning, "Generating typed dependency parses from phrase structure trees," in *LREC*, 2006.
- [12] K. C. Litkowski, *Syntactic Clues and Lexical Resources in Question-Answering*, 2001.
- [13] L. Qiu, M. yen Kan, and T. seng Chua, "A public reference implementation of the rap anaphora resolution algorithm," in *LREC*, 2004.
- [14] S. Banerjee and S. Banerjee, "An adapted lesk algorithm for word sense disambiguation using wordnet," in *CICLing*, 2002.
- [15] M. M. Stark and R. F. Riesenfeld, "Wordnet: An electronic lexical database," in *Eurographics Workshop on Rendering*. MIT Press, 1998.
- [16] S. Helmreich and D. Farwell, "Text meaning representation as a basis for representation of text interpretation," in *Conference of the Association for Machine Translation in the Americas on Envisioning Machine Translation in the Information Future*, 2000.
- [17] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007.
- [18] S. Ray and R. H. Turi, "Determination of number of clusters in k-means clustering and application in colour image segmentation," *ICAPRDT*, 1999.
- [19] T. Kamps, J. Kleinz, and J. Read, "Constraint-based spring-model algorithm for graph layout," in *Proceedings of the Symposium on Graph Drawing*. Springer-Verlag, 1996.
- [20] BBC, "BBC Historical Figures," http://www.bbc.co.uk/history/historic_figures/, 2012, [Online; accessed January-2012].