Document Concept Hierarchy Generation by Extracting Semantic Tree Using Knowledge Graph

Conference Paper · December 2018

DOI: 10.1109/WIECON-ECE.2018.8783083

2 authors:

Sanjida Nasreen Tumpa
The University of Calgary
21 PUBLICATIONS **13** CITATIONS

SEE PROFILE

Muhammad Masroor Ali
Bangladesh University of Engineering and Technology
19 PUBLICATIONS **60** CITATIONS

SEE PROFILE

Document Concept Hierarchy Generation by Extracting Semantic Tree Using Knowledge Graph

Sanjida Nasreen Tumpa[1, 2] and Muhammad Masroor Ali[1, 3]
[1]Department of CSE, Bangladesh University of Engineering and Technology (BUET), Dhaka, Bangladesh
[2]tumpa.sanjida@gmail.com, [3]mmasroorali@cse.buet.ac.bd

***Abstract*—Semantic Web, as an extension of traditional web, is concerned about the vast amount of unstructured data, and with its motive to make the entire knowledge content machine readable, as well as machine interpretable, all the processes of structuring the data is highly significant. Knowledge representation in trees has been a familiar mechanism for some time. However, such representations lack in existence when it comes to document content. In this paper, we present a general mechanism that can generate a representation of the concepts of any document in the form of knowledge trees. We further gather knowledge from knowledge graphs and analyze these data by mapping it with an existing ontology. Finally, we explain how this can be used to create hierarchical concept recommendations to make the documents search efficient.**

***Index Terms*—Concept hierarchy, Content tree, Information retrieval, Semantic web, Text mining**

I. Introduction

In this golden era of information technology, Semantic Web [1]–[3] plays a vital role by establishing well defined expressions to extract meaningful pattern and information. Most of the information in traditional web is unstructured and not suitable for machines to retrieve intelligently. Semantic web focuses on structuring the immeasurable amount of information to make these information machine readable. As a result, analysis of tonnes of documents which seemed nearly impossible previously can now be interpreted by machines.

To extract knowledge from documents, the concept of text mining was introduced. Text mining [4] is the science of retrieving high quality information from unstructured documents and transform them into structured format for further analysis. Linking and retrieving structured information are also required to establish the concepts of Semantic Web. Thus, the concepts of knowledge graph and ontology possess their own significance in the field data linking. If the contents of documents can be represented in a structured way, it will be possible to link core topics with existing knowledge sources. Also, this will help to integrate those into the existing taxonomy to categorize the document in order to make the searching process efficient.

This paper proposes a novel approach to generate document content tree that represents the texts in the document in a way that the semantic relations are preserved. The approach also finds out the topics or concepts from the documents provided. This paper further intends to determine the hierarchical structure of core concepts by integrating the knowledge graphs with the content tree.

The rest of this paper is organized as follows. Section II provides the overview of the relevant research works. In Section III, our motivation is presented, which led us to work in this area. Section IV and Section V give the preliminary knowledge followed by the conceptual design of our approach. Section VI demonstrates the experimental result of our system. Finally, Section VII concludes the paper.

II. Literature Review

Many researchers worked on the document content representation for text mining or information retrieval [5]–[11]. Some of them considered the semantics of documents, some did not pay any heed to it.

The classical method of information retrieval, Boolean model, focused only on the presence of any word in the document without considering the semantic relations [5]. A popular method for object categorization, the bag-of-words model, disregarded the semantic relation and order of words. However, it considers multiplicity [6]. Another notable one, Vector Space Model, reduced the limitation of binary weights by representing each document in a vector space.However, the semantics of words was also not preserved [7]. Some researchers worked on the extended versions of the above models to incorporate the semantic relations along with term representation [8], [9], [11].

To incorporate text structure and context with document content representation, researchers have proposed some graphical representation based approaches. In [12], a graph based text representation method was designed under word semantic space to obtain parts of speech, order, frequency, co-occurrence and context of words in the document. Apart from this, a graph based text mining technique, GDClust has been proposed in [13] based on co-occurrence of frequent senses to present text document as hierarchical document graphs. In [14], a semantics based graph structure was proposed to hold more structural information and mutual semantic relationship among words. In [15], a term graph model was proposed to represent the content and relationship among words. In [16], a conceptual graph representation of text was proposed using existing linguistic resources, verb net and word net.

In [17], a method for extracting document fragment automatically from structured online documents with internal hierarchical structure like HTML, XML, SGML etc. was discussed. In [18], sentence tree structure for document summarization was used. The authors did not represent document content as a rooted tree.

Besides enlightening the representation of document contents, we focused on the hierarchical representation of concepts. In [19], a hierarchical organization of concepts from a set of documents was proposed. The authors preferred to use subsumption to create the hierarchy of selected terms. In [20], an approach for concept hierarchy using formal concept analysis was proposed. This approach is based on distributional hypothesis, and the hierarchy between terms have been decided considering syntactic dependencies.

There are some other works related to text representation and concept hierarchy, except the mentioned researches. However, no researcher focused on the collection of more knowledge regarding the core concepts of a document from the knowledge bases to cluster the concepts and also to get the document hierarchy. Furthermore, from the above discussion, it can be said that the concept of document content tree is a barely touched topic. Trees are used in the taxonomic representations of concepts, but according to semantic relation and context, it is not in the field of document content representation. This paper intends to contribute in such gaps of information retrieval and text mining.

III. Motivation

Content tree itself is quite significant for information retrieval and text mining. When considered any document containing ample amount of text, it may be possible for humans to understand the concepts, but it will not be possible for machines to understand them. This particular requirement is getting universal attention as the present world focuses on parsing and analyzing data in the minimum unit of time.

Furthermore, establishing connections or links between knowledge from different sources is very much significant in Semantic Web. If the knowledge can be represented in well-defined data structures, say tree, for example, establishing such connections will become so much easier. Using a tree to represent the concepts will contribute even more as it can represent data in hierarchical form and thus, machines will be able to discard unnecessary data when working on a specific operation, improving the search time and lowering the operational complexity. Positive effects will be viewed in various aspects, mainly in case of knowledge clustering. Considering all these pros, we were highly motivated to work on knowledge representation using content tree.

IV. Document content tree and Concept hierarchy

A. Document Content Tree

Document content tree can be defined as a tree based representation of any document, that demonstrates the dependencies among words in that document. A content tree is basically an acyclic directed graph, $G = \{N, E\}$ where N and E are the set of nodes and edges respectively.

In the content tree model proposed in this paper, every document is converted into a rooted tree based on the concepts present in it. There can be multiple trees or a forest, if the document discusses about multiple topics. Content tree has three types of elements: root, nodes and edges. The general structure of a document content tree is as follows:

- **Root:** A tree contains the information regarding the root node only. Root has been chosen from the main entities of the sentences in the provided document.
- **Nodes:** Nodes denote the concepts of the document. Nodes can consist of single or multiple words.
- **Edges:** A directed edge between two nodes resembles the relation between the nodes. Usually, verbs of the sentences are chosen as the edges as verbs represent the relations in the sentences.

B. Concept Hierarchy

In Natural Language Processing, concepts can be expressed as senses of a document. A single concept may consist of a single word or can have multiple words. Concept hierarchy establishes the hierarchical structure in this scenario. Concept hierarchy or taxonomy is a mechanism to demonstrate the generalized hierarchical relationships among concepts. It ensures efficient categorization of a document.

V. Conceptual Overview of the System

The overall system can be dichotomized into two major parts:

1) Document content tree generation,
2) Concept hierarchy extraction.

A. Document Content Tree Generation

The generation of content tree varies with the language of the document. The tree generation module comprises of the following steps:

1) **Tokenization and Preparing Tagged Document**: The document is segmented based on some pre-defined separators. At this stage, we need to preserve some information together, like name, date etc. For instance, if we use white space as separator, the name "সাকিব আল-হাসান" ([sakib al-fiasan], Sakib Al-Hasan) will be tokenized into two tokens as "সাকিব" ([sakib], Sakib) and "আল-হাসান" ([al-fiasan], Al-Hasan) which is not anticipated. For this reason, external knowledge is incorporated to get the desired tokens. We also extract information related to a token. Some examples are the sentence position of that particular token in the provided document, token position in a sentence and overall token position in the document.
2) **Coreference Resolution**: A combination of heuristic and supervised methods is used for coreference resolution [21]. It helps to find all expressions that refer to the same entity. Therefore, our method recursively tracks the possible antecedent and the pronoun is replaced by the referred entity.
3) **Labeling and Filtration of Extracted Information**: An extensive dictionary is used to determine the parts of speech along with some additional knowledge regarding the extracted tokens. For example, the parts of speech of "বাংলাদেশ" ([baŋladeʃ], Bangladesh) is *Noun* but more specifically, it is the name of a *Country*. Furthermore, not

all tokens hold significance for the content tree. Thus, we use Bengali dictionaries for the less significant words, in order to filter the tokens.

4) **Tree Construction**: We construct the set of nodes and edges after filtration. Every edge in E is considered as a directed edge between two vertices N_i and N_j. The input of the tree algorithm is the extracted information and the output is the representation of content tree.
5) **Tree Optimization**: We merge some nodes to optimize the content tree that results in the reduction of traversal time. Therefore, we propose some semantic rules to unify adjectives with noun, adjectives with adjectives etc. and we consider the following notations to establish the theoretical terms for the rules: *CurrNode* is the present node, *PrevNode* is the immediate preceding node of the present one in the tree and *ResultNode* is the output node after applying a rule. E_1 is the edge between the previous node of *PrevNode* and *PrevNode* itself. E_2 is the edge between *PrevNode* and *CurrNode*. The semantic rules are as follows:
   - **Rule #1:**
     IF *CurrNode* $\rightarrow$ Noun $\wedge$ *PrevNode* $\rightarrow$ Noun
     THEN
     ResultNode = MERGE (*PrevNode*, *CurrNode*),
     ResultEdge = MERGE (E_1, E_2),
     POS_OF (*ResultNode*) = POS_OF (*CurrNode*)
   - **Rule #2:**
     IF *CurrNode* $\rightarrow$ Noun $\wedge$ *PrevNode* $\rightarrow$ Adjective
     THEN
     ResultNode = MERGE (*PrevNode*, *CurrNode*),
     ResultEdge = MERGE (E_1, E_2),
     POS_OF (*ResultNode*) = POS_OF (*CurrNode*)
   - **Rule #3:**
     IF *CurrNode* $\rightarrow$ Month $\wedge$ *PrevNode* $\rightarrow$ Number
     THEN
     ResultNode = MERGE (*PrevNode*, *CurrNode*),
     ResultEdge = MERGE (E_1, E_2),
     POS_OF (*ResultNode*) = Date
   - **Rule #4:**
     IF *CurrNode* $\rightarrow$ Adjective $\wedge$ *PrevNode* $\rightarrow$ Adjective
     THEN
     ResultNode = MERGE (*PrevNode*, *CurrNode*),
     ResultEdge = MERGE (E_1, E_2),
     POS_OF (*ResultNode*) = Adjective

B. Document Concept Hierarchy Using Knowledge Bases

Concept hierarchy increases efficiency of document retrieving in a large scale. To implement the concept hierarchy, we maintain the following steps:

1) **Concept Extraction from Provided Document:** The above mentioning document content tree helps us to extract the major concepts of a document. Here, we consider the roots of the trees in the set of core concepts.
2) **Knowledge Extraction Using Knowledge Bases:** After extracting concepts from the document, existing knowledge bases, like Google Knowledge Graph [22], DBpedia [23], YAGO [24], WordNet [25] etc. are used to gather more information. Knowledge graphs basically provide knowledge as linked data. The yielded information then requires to embed to be more meaningful.
3) **Word Embedding and Similarity Checking:** The process of word embedding demonstrates a class of approaches for representing words in a continuous vector space where semantically similar words are mapped to nearby points [26]. There are two popular methods of word embedding from text, namely Word2Vec [27] and GloVe [28].
   We incorporated Word2Vec method to embed the information in the vector space to determine the similarity among the information extracted from the knowledge bases.
4) **Concept Clustering and Hierarchy Generation:** The extracted information is clustered based on the similarity weight. There are pre-defined cluster tags for all clusters which are obtained from DBpedia Ontology. The taxonomic representation of extracted information is generated following the class hierarchy of this particular ontology.

VI. Experimental Result

We implemented the system using JAVA programming language. Though we have conducted our experiment on Bengali documents, the proposed approach will work similarly for English documents also. We have used some modified version of Wikipedia pages eliminating the complex sentences as input. After pre-processing the document, we used SQLite database to keep the tagged tokens along with all information. A sample tagged sentence is like: "সাকিব <Noun, Person, Male> আল-হাসান <Noun, Person> একজন <Noun, Number> বাংলাদেশী <Adjective, Nationality> ক্রিকেটার <Noun, Profession>" ([sakib al-ɦasan ɛkdɜ baŋladeʃi kriketar], Sakib Al-Hasan is a Bangladeshi cricketer). After that, we constructed the tree following the proposed algorithm. Then, we extracted information from existing knowledge bases using the roots of the content tree. As of now, we have used Google knowledge graph and DBpedia for this purpose. Then, we have used DBpedia ontology to cluster the data in order to create the concept hierarchy. Figure 1 demonstrates the resultant content tree along with concept hierarchy. The accuracy of our system for simple sentences are quite satisfactory. It could identify 97-98 nodes among 100 nodes.

VII. Conclusion

Document hierarchy based on content tree will contribute predominantly in the taxonomy of documents. The tree representation will also help us to merge any document with the same concepts to increase connectivity of knowledge efficiently. In addition, this will open up a vast field of research area to represent the documents in a more structural manner, making it more search efficient and ultimately achieving the vision of Semantic web.

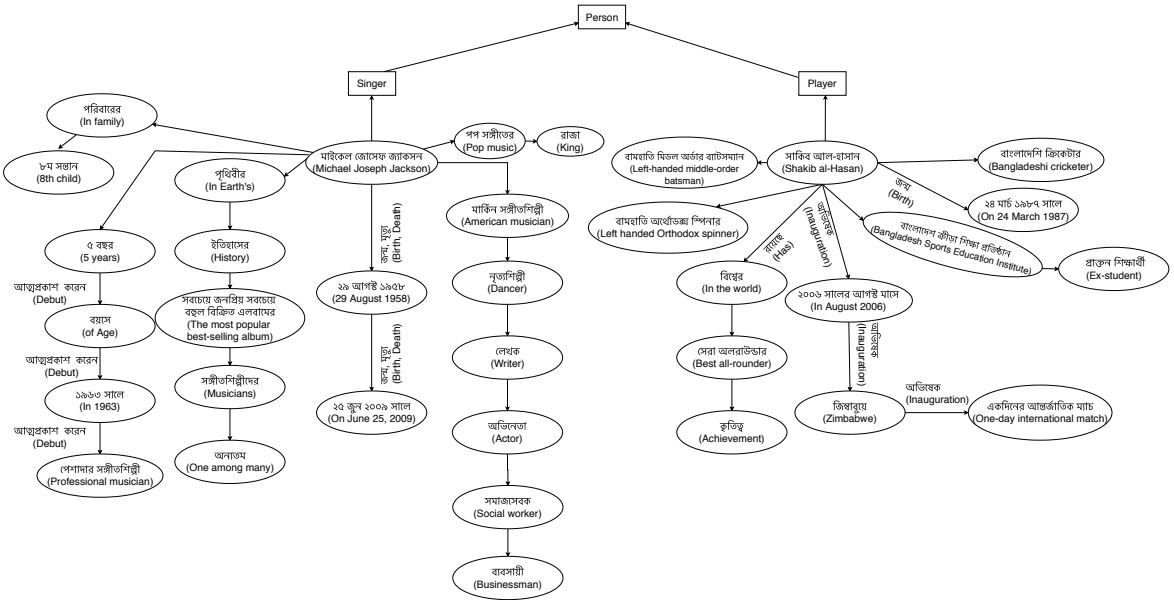

Figure 1. Document content tree generated from the document on "Sakib Al-Hasan" and "Michael Joseph Jackson".

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 34–43, 2001.

[2] P. Hitzler, M. Krotzsch, and S. Rudolph, *Foundations of semantic web technologies*. CRC Press, 2009.

[3] L. Yu, *A developer's guide to the semantic Web*. Springer Science & Business Media, 2011.

[4] A.-H. Tan *et al.*, "Text mining: The state of the art and the challenges," in *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*, vol. 8. sn, 1999, pp. 65–70.

[5] A. H. Lashkari, F. Mahdavi, and V. Ghomi, "A boolean model in information retrieval for search engines," in *Information Management and Engineering, 2009. ICIME'09. International Conference on*. IEEE, 2009, pp. 385–389.

[6] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1, no. 1-4, pp. 43–52, 2010.

[7] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[8] P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser, "Latent semantic analysis," in *Proceedings of the 16th international joint conference on Artificial intelligence*. Citeseer, 2004, pp. 1–14.

[9] G. Salton, E. A. Fox, and H. Wu, "Extended boolean information retrieval," *Communications of the ACM*, vol. 26, no. 11, pp. 1022–1036, 1983.

[10] W. Waller and D. H. Kraft, "A mathematical model of a weighted boolean retrieval system," *Information Processing & Management*, vol. 15, no. 5, pp. 235–245, 1979.

[11] E. A. Fox, "Extending the boolean and vector space models of information retrieval with p-norm queries and multiple concept types," Ph.D. dissertation, Cornell University, Ithaca, NY, USA, 1983.

[12] F. Zhou, F. Zhang, and B. Yang, "Graph-based text representation model and its realization," in *Natural Language Processing and Knowledge Engineering (NLP-KE), 2010 International Conference on*. IEEE, 2010, pp. 1–8.

[13] M. S. Hossain and R. A. Angryk, "Gdclust: A graph-based document clustering technique," in *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 417–422.

[14] J. Wu, Z. Xuan, and D. Pan, "Enhancing text representation for classification tasks with semantic graph structures," *International Journal of Innovative Computing, Information and Control (ICIC)*, vol. 7, no. 5, 2011.

[15] W. Wang, D. B. Do, and X. Lin, "Term graph model for text classification," in *International Conference on Advanced Data Mining and Applications*. Springer, 2005, pp. 19–30.

[16] S. Hensman, "Construction of conceptual graph representation of texts," in *Proceedings of the Student Research Workshop at HLT-NAACL 2004*. Association for Computational Linguistics, 2004, pp. 49–54.

[17] V. Maslov, "Method for extracting digests, reformatting, and automatic monitoring of structured online documents based on visual programming of document tree navigation and transformation," Mar. 25 2003, uS Patent 6,538,673.

[18] Y. Kikuchi, T. Hirao, H. Takamura, M. Okumura, and M. Nagata, "Single document summarization based on nested tree structure," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, 2014, pp. 315–320.

[19] M. Sanderson and B. Croft, "Deriving concept hierarchies from text," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 206–213.

[20] P. Cimiano, A. Hotho, and S. Staab, "Clustering concept hierarchies from text," in *Proceedings of the Conference on Lexical Resources and Evaluation (LREC)*, 2004.

[21] W. M. Soon, H. T. Ng, and D. C. Y. Lim, "A machine learning approach to coreference resolution of noun phrases," *Computational linguistics*, vol. 27, no. 4, pp. 521–544, 2001.

[22] "Google knowledge graph api," [Online; accessed 27-July-2018]. [Online]. Available: https://developers.google.com/knowledge-graph/

[23] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*. Springer, 2007, pp. 722–735.

[24] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A large ontology from wikipedia and wordnet," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 3, pp. 203–217, 2008.

[25] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[26] "Vector representations of words," 2018, [Online; accessed 3-August-2018]. [Online]. Available: https://www.tensorflow.org/tutorials/representation/word2vec

[27] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, 2014.

[28] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.