Visualizing UML’s Sequence and Class Diagrams Using Graph-Based Clusters

Nakul Sharma
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram,Guntur-522502, India
nakul777@gmail.com

Prasanth Yalla
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram,Guntur-522502, India
prasanthylla@kluniversity.in

Abstract. The paper discusses the creation of UML diagram based recommendation system using java and class files as the input. The existing systems do not make use of techniques available in text-mining for creating UML diagrams. The overall methodology makes use of keyphrase extraction, contextual similarity calculation, and graph-based clusters in creating UML diagrams. The existing systems survey of state-of-art UML diagram generation techniques and keyphrase extraction survey is also provided. A comparative analysis of the existing tools for generating UML-diagrams is also provided. The recommendation system generated is useful to maintenance engineers and software developers.

Keywords: Natural Language Processing, UML diagrams, Sequence Diagram, Source Code Analysis, Keyphrase Extraction, Clustering.

1. Introduction
UML diagram is considered as the industrial standard for providing visual representation in the design phase of SDLC. These diagrams are part of the Software Design Document which is a work product of the design phase. There are two categorizations of UML diagrams, namely the static or structural specification and the behavioral or dynamic specification [32].

In the current work, a literature review of UML diagram construction from text or source code is done. A comparative analysis of different methods used in UML diagram construction is also proposed. The
authors also highlight UML diagrams as a recommendation system for developers as well as the maintenance engineer. The UML diagrams can give a visual insight into the effect of the system’s changes as they are incorporated by the developer and the maintenance engineer [3]. In the existing literature, natural Language text is used in developing UML diagrams. These techniques however, cannot be applied to large documentations or source codes. This is because poor exploration of the contextual relationships among the source code and API documents.

Saroj and Monali proposed an unsupervised graph based keyword extraction method called Keyword Extraction using Collective Node Weight (KECNW) which determines the relevance of keyword by taking into cognizance various collective parameters. The authors also propose doing preprocessing and providing textual graph representation [63]. These systems are not suitable for large open source code projects because of high computational memory and time.

Aziz et. al. have proposed Extended ForUML which generates UML sequence & class diagram from Fortran source code. The technique applied is unsuitable for large open source projects since computational time & memory requirements are not met [69].

Mohammed et. al. proposed clustering technique by considering the knowledge of artifacts. However, authors proposed work cannot identify software architecture efficiently for large software projects [34].

In this paper, the authors propose a contextual similarity approach combined with cluster and graph creation. A multi-step approach involves keyphrase extraction, graph construction, clustering of related documents together and finally creation of UML class & sequence diagrams.

The rest of the paper is divided into the following sections: Section 2 discusses the related literature. Section 3 gives the proposed methodology while section 4 gives the result & discussion. Lastly, section 5 gives future scope & conclusion.

2. Literature Review
In the current section, a literature review based on UML diagrams generation, keyphrase extraction, and clustering are discussed. Table-1 gives a comparative analysis of research papers related to UML diagram construction.

| Name of Tool Generated | Advantages of Tool | Disadvantages of Tool |
|------------------------|-------------------|----------------------|
| Extended ForUML (2019) | 1. Extended the ForUML system to include sequence diagrams generation 2. More efficient and useful in different scenarios. | Except for class and sequence diagram, other UML diagrams are not generated. |
| Automatic Builder of Class Diagram (ABCD) (2016) | The best properties of NLP techniques are employed in generating UML Class Diagrams. | 1. The system lacks advanced mechanisms for checking repetitive information 2. Two synonyms were treated differently in the case of the class diagram. |
| RECAA (2015) | 1. Provides an automated approach towards the development of the UML class diagram. | The UML diagrams such as activity, state-chart are not drawn. |
ForUML (2015)  
1. ForUML can reverse engineer FORTRAN code.  
2. ForUML is able to convert FORTRAN into UML code.  
3. ForUML combines AgroUML in converting class diagram  
   plugin

Class-Gen (2010)  
Raw Evaluation and comparative analysis of the system (Class-Gen) developed was done.  
Aggregation and inheritance not included in the system developed.

UMGAR (2008)  
Automatic identification of OO Concepts  
Not very efficient in generating state chart and class diagram

ER convertor (2008)  
The tool used heuristics to convert English language  
Does not propose a semantic heuristic

SENSE (2007)  
1. Meta Model elements can be combined  
2. Appropriate semantic encode  
The application of pattern is difficult Tool does not study the syntactic heuristics

LIDA (2001)  
Generates class diagrams  
The tool is not fully automated.

CM-Builderr (2000)  
Generates class diagrams.  
Manual refinement not yet stable.

Table 2 Summary Of Literature Review on Developing UML Diagrams.

| Title of Publication | Work Done/Conclusion of Paper | Future Work Purposed | References |
|----------------------|-------------------------------|----------------------|------------|
| Minimizing ambiguity in Natural Language Software Requirements Specification | Complete analysis of NL text to create UML Diagram artifacts | 1. Improving accuracy by improving algorithm  
2. Component and Deployment diagram not generated | [4] |
| From Natural Language Software Specifications to UML Class Models | Provide better accuracy for Conversion of NL to Class Models | OCL for natural language constraint | [5] |
| From User Requirements to UML Diagrams | Ontology and NLP techniques for developing class diagram. ANNIE Tool developed | 1. Heuristics rules are not exhaustive  
2. Algorithm for text analysis can be improved in order to generate complex UML Diagrams | [7] |
| Generating class models through controlled requirements | 1. Overcoming ambiguity by making the language controlled  
2. Generate effective, quality usecases (RAVEN) tool developed. | Benchmark for requirement validation is necessary part of future | [8] |
| UDRA: Reflecting Natural RACE tool generated | Not | | [10] |
| Language Text into UML Diagrams | For converting NL text to UML Diagram. The diagrams generated were class, object, deployment, package diagrams |  |
|--------------------------------|-------------------------------------------------------------------------------------------------|--|
| Automatically generating object models from N.L. Analysis | Original problem statements are converted to 4w language. GOOAL tool made to complete the task of conversion | [11] |
| A CBR approach to Text to Class Diagram Translation | REBUILDER UML developed for generating class diagram. Ontology used is domain ontology for conversion. System should be evaluated by software engineers | [25] |
| A linguistic approach to the development of object-oriented system using the NL system LOLITA | OO Model extracted using NLP systems. Implement the missing functionality & experiment with real life requirement documents | [23] |
| CM-Builder A Natural Language-based CASE Tool | A Case tool CM-Builder is used for undertaking conversion of text to UML Class diagram | 1. Building Language corpus for real world applications 2. More grammar rules for the sentence types 3. Discourse interpretation module improved 4. Extraction dynamic aspects from NL Text 5. Evaluation can improved |
| UCDA: Use Case Driven Development Assistant Tool for Class Model Generation | UCDA for generating use case diagram was developed. Tool has advantages of reducing time to market, increasing productivity and helping novice developers | Automation can help software design effort and cost getting reduced |
| The MOVA Tool: A Rewriting Based UML modeling, measuring, validation tool | The MOVA tool is rewriting based UML tool developed with the aim of integrating SE software engineering curricula and OCL expressions cannot be introduced with MOVA, so it is a future work | [26] | [27] |
| Relative Extraction Methodology for class diagram generating using dependency graph | Generating UML diagrams from natural language specifications | Relative Extraction methodology implemented Data structure’s concepts such as BFS and DFS used for generating UML Diagrams | Multiplicity between classes constitute future work |
|---|---|---|---|
| RAPID architecture given for the purpose of generating UML diagrams from NL Text | From Requirements to UML Models & Back : How automatic processing of text can support requirements engineering | Aims for constructing requirement engineering specification with consistent models. RECAA tool that automates UML diagram generation is developed. | Not mentioned |
| Automatic Builder of Class Diagram (ABCD) : An application of UML generation from Functional requirements | Generated Use Case description as a basis for Object Oriented Class Model generation | Utilized statistical and pattern recognition techniques in generation of UML class diagrams. 1. System lacks advanced mechanism to deal with redundant information problem. 2. A deep NLP analysis can help in avoiding association method confusion. | |
| Created Class-Gen for creating UML Diagrams from natural language text. Potential exists in utilizing NLP tools for software development | Conceptual Modeling of Natural language functional requirements | Claims that automation can help save time of developers. | None |

There are several methodologies being used in developing UML diagrams. The most common diagrams which are developed are Use –case and class diagram[13][15][16][18] [30]. However little work has been done wrt using text, source code, API documentation for generating UML diagrams. In addition text mining techniques are not used extensively in generating UML diagrams.
| Reference number | Authors names | Title of Paper | Advantages of Proposed Work | Disadvantage of Proposed Work |
|------------------|---------------|----------------|-----------------------------|-------------------------------|
| 34               | G S. Mohammadi and H. Izadkhah | A new algorithm for software clustering considering the knowledge of dependency between artifacts in the source code | Similarity Index along with various hierarchical and search based approaches have been used | Authors proposed work cannot identify software architecture efficiently for large software projects. |
| 35               | V. U. Gómez, S. Ducasse and T. D’Hondta | Visually characterizing source code changes | Authors propose Visual representation of changes done on source code. text and visual representations are combined. | Semantic sequences are not addressed properly for large source code projects. Release masters are also not addressed. |
| 36               | S. L. Abebe and P. Tonella | Extraction of domain concepts from the source code | Concepts of NLP ontologies mapped onto source code by the authors. Union of ontologies give better result. | None Mentioned |
| 37               | S. Bajracharya, J. Ossher and C. Lopes | Sourcerer: An infrastructure for large-scale collection and analysis of open-source code | The authors develop sourcerer which provides structural as well as textual medium for searching. | Software Engineering Tools integration needed. |
| 38               | Abihoa, Willyan D., and Leandro N. De Castro. | A keyword extraction method from twitter messages represented as graphs. | Authors concluded that the graph developed using neighborhood edges provided better performance. | Other centrality measures can also be applied. |
| 39               | Ziqi Zhang, Johann Petrak, Diana Maynard, | Adapted TextRank for Term Extraction: A Generic Method of Improving Automatic Term Extraction Algorithms | A generic method for term extraction proposed. | For pre-training in the seeding stage other techniques can also be used. |
| 40               | Duari, Swagata, and Vasudha Bhatnagar. | Complex Network Based Supervised Keyword Extractor. | Text is considered as a complex network. The proposed methodology is domain ad language independent. | Work can be extended to different Indian languages as well. |
| 41               | Wael Etaibwi, Arafat Awajan | Graph-based Arabic text semantic representation | Semantic relationships found for constructing AG, can be used in text summarization for different Indian languages can be constructed. | AG for different Indian languages can be constructed. |
| 42               | Javad Rafiei-Asl, Ahmad Nickabadi | TSAKE: A topical and structural automatic keyphrase extractor | The main aim of keyphrase extraction is fulfilled | Macro & micro topic selection forms the future work. |
| 43               | Shansong Yang, Weiming Lu, Dezhi Yang, Xi Li, Chao Wu, Baogang Wei | KeyphraseDS: Automatic generation of survey by exploiting keyphrase information | Document summarization tool which overcomes the data sparsity problem. | Multiple layers ML/DL algorithm can be applied. |
| 44               | B. D. Farahani, S. O. Fatemi and M. Ghorbani | Automatic Keyphrase Extraction from Persian Scientific Documents Using Semantic Relations | Due to lack of linguistic resources in Persia authors make use of thesaurus for conducting keyphrase extraction in Persian language. | Keyphrase suggestion, conducting manual evaluation. |
| 46               | Niraj Kumar, Kannan Srinathan, Vasudeva Varma | Towards Intelligent Text Mining Under Limited Linguistic Resources | All these techniques make use of graph for representing text. | Not suitable for large text documents. |
| 47               | Litvak, M., Last, M. & Kandel, A. | DegExt: a language-independent keyphrase | Performance is better than TextRank | Additional natural languages can be |
The existing literature also has several survey papers on the topic of Keyphrase extraction [51-57]. Yakoob et al proposed keyword searching in the cloud environment. The author gave the strategies for retrieving essential information from cloud records. The authors work is novel as keyword as well as keyphrase locations are assessed in cloud environment [49]. Selvaraj et. al propose a new relationship for extracting essential words from a given document. Lexical association between words is used for finding relevant keywords. The authors propose an approach to vertex connectivity for identifying different centrality measures of keywords [62].

3. Proposed Methodology

In the existing systems, centrality measures have been applied to data structures and graph-based systems as well for clustering or for keyphrase extraction. Software Engineering research inculcates application from variety of domains [68-69]. This trend of applicability is also see with NLP domain [68]. An extensive analysis of clustering algorithm is presented by the authors [45] [64] [67]. The authors proposed making use of contextual similarity in generating UML diagrams. This first step includes extracting the methods and fields from the source code and class files. Using the keyphrases extracted a Source code directed graph is constructed. The graph is used in calculating similarity index between two terms.
The proposed methodology is divided into following sub-sections:

3.1. Module-1 Probabilistic Weighted based contextual similarity measure for Source code and class files dependency graph [33]

The source code’s metrics and class-file metrics are used in developing SDG graph. Here vertex V is source code methods and fields while the edges represent the weighted rank. The edge weights are calculated based on the probability of occurrence of methods and fields. The last step includes generating Contextual Source Code Dependency Graph similarity index [33].

3.2. Module-2 Contextual source code graph based clustering algorithm for class dependency diagram [33]

The similarity index computed in the previous module is now used in clustering related files together. The cluster of files indicates the relative relatedness as it exists according to similarity index. These clusters of files are used to construct the Class and Sequence diagram [33].

4. Results & Discussion

The project was tested on weka software. The project accepts three types of inputs, the first is directory location where the source code files are located. Figure-2 and Figure 3 shows the input provided for source code path and class path respectively.

Figure 2: Input for Source Code Path
The input of all the essential locations gets completed and categorization in form of tokens, for each word of the document gets started. As shown in figure-4, a snapshot of processing at tokenization stage gives information about different token in the file name ConsoleLogger.java.

The source code and its documentation are scanned leading to the development of the graphs. The similarity measure is then used in the generation of clusters. These clusters are then used in the generation of UML’s class diagram and sequence diagram.
Figure 5: Accepting Input for Number of Iterations

Figure 6: Accepting Input for Number of Clusters
The current work can be seen as developing UML and sequence diagrams by making use of contextual similarity, clusters and keyphrase extraction from the source code and API documentation. The traceability link across the different artifacts within the SDLC holds the key to providing a better maintenance environment.
Table 4. Comparative Analysis of Different UML Tools Developed

| Name of UML Tool | Techniques and Input files used for Conversion | NLP SOFTWARES | NLP AND Rules (Heuristics) | NLP AND XMI/XML | Source Code | API Documentation |
|------------------|-----------------------------------------------|---------------|-----------------------------|------------------|-------------|-------------------|
| Automatic Builder of Class Diagram (2016) | | No | Yes | Yes | No | Yes |
| RECAA (2015) | | Yes | Yes | No | No | Yes |
| CM-Builder (2000) | | Yes | Yes | No | No | No |
| UMGAR (2008) | | Yes | Yes | No | No | No |
| SENSE (2007) | | Yes | Yes | No | No | No |
| ER convertor (2008) | | No | Yes | No | No | No |
| LIDA (2001) | | No | Yes | No | No | No |
| FortUML (2015) | | Yes | Yes | Yes | Yes | No |
| Extended ForUML (2019) | | Yes | Yes | Yes | Yes | No |
| Nakul Sharma et al [33] | | Yes | Yes | No | Yes | Yes |

Table 5. Comparative Analysis of Proposed Work with Extended ForUML [69].

| Sr. No. | Own Work | Extended ForUML [69] |
|---------|----------|----------------------|
| 1 | Generate class and sequence diagram using API documents and source code as input | Generate sequence and class diagram using fortron code only. |
| 2 | Parsing of all java source code for names of methods and fields | Parsing of fortron source code done using Open Fortron Parser (OFP) available for Fortron Programming Language. |
| 3 | Creation of dependency graphs | Parsing of Fortron source code done using Open Fortron Parser (OFP) library. |
| 4 | Calculation of contextual similarity for the source code, class files and API documentation | Development of XMI representation using derived relationship between code chunks. |
5. Conclusion & Future Scope

The paper discusses how UML diagram can be used as a tool for recommending most essential classes within a given set of project. A large-scale open source project cannot be assessed using the existing similarity measures. Hence, a new hybrid probabilistic model is proposed for large open-source projects. The future scope of current work includes:

1. Construction of static and dynamic UML diagrams from Source Code, Natural Language artifacts.
2. Creating Recommendation System for different stakeholders within the SDLC.
3. Developing specialized ontologies for developers, coders, designers, UI experts to enable lesser time to market.
4. Developing traceability mechanism between the different software artifacts.
5. Centrality measures can be used to further recommendation system research.

References

[1] Pankaj Jalote. 2002. An Integrated Approach to Software Engineering, Second Edition, Narosa Publishing House.
[2] Bernd Bruegge, Allen H. Dutoit. 2012. Object Oriented Software Engineering Using UML, Patterns, and Java, Pearson Education, Tenth Edition ISBN: 978-81-775-8768-5.
[3] Dan Pilone, Neil Pitman. 2007. UML 2.0 In a Nutshell, O'Reilly Media, Third Edition. ISBN: 81-8404-002-4.
[4] A. Umber and I. S. Bajwa, "Minimizing ambiguity in natural language software requirements specification," 2011 Sixth International Conference on Digital Information Management, Melbourn, QLD, 2011, pp. 102-107, doi: 10.1109/ICDIM.2011.6093363.
[5] Bajwa I.S., Choudhary M.A. (2012) From Natural Language Software Specifications to UML
Class Models. In: Zhang R., Zhang J., Zhang Z., Filipe J., Cordeiro J. (eds) Enterprise Information Systems. ICEIS 2011. Lecture Notes in Business Information Processing, vol 102. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-29958-2_15.

[6] Pushpak Bhattacharyya. 2012, Natural Language Processing: A perspective from Computation in Presence of Ambiguity, Resource Constraint and Multiliguality, In proceedings of the CSI Journal of Computing, 1, 2, (2012).

[7] Herchi, Hatem, and Wahiba Ben Abdessalem. "From user requirements to UML class diagram." arXiv preprint arXiv:1211.0713 (2012).

[8] Giganto, Reynaldo. "Generating class models through controlled requirements." NZCSRSC-08, New Zealand Computer Science Research Student Conference Christchurch, New Zealand. 2008.

[9] Mathias Landhauber, Sven J. Korner and Walter F. Tichy. 2014, From Requirements to UML Models and Back: how automatic processing of text can support requirements engineering. In. Proc. Software Quality General, Springer, March 2014, Vol 22, Issue 1, pp 121-149.

[10] Ontology, Domain. "UDRA: Reflecting Natural Language Text in to UML Diagrams."

[11] H. G. Pérez-González, Jugal K. Kalita, Alberto Salvador Núñez Varela, and Richard S. Wiener. 2005. GOOAL: an educational object oriented analysis laboratory. In Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications (OOPSLA ’05). Association for Computing Machinery, New York, NY, USA, 180–181. DOI:https://doi.org/10.1145/1094855.1094924

[12] Mencl V. 2004, Deriving Behavior Specifications from Textual Use Cases, In. Proc. of Workshop on Intelligent Technologies for Software Engineering (WITSE04, Sep 21, 2004, part of ASE 2004), Linz, Austria, ISBN 3-85403-180-7, pp. 331-341, Oesterreichische Computer Gesellschaft, September 2004.

[13] Imran Sarwar Bajwa and M. Abbas Choudhary. 2006, Natural Language Processing based automated system for UML diagrams generation In. Proc. 18th National Computer Conference (NCC-2006), Page No-1-6.

[14] Krishnan, Hema, and Philip Samuel. "Relative Extraction Methodology for class diagram generation using dependency graph." 2010 International Conference On Communication Control And Computing Technologies. IEEE, 2010.

[15] Hina Afreen Imran Sarwar Bajwa and Behzad Bordbar, 2011. “SBVR2UML: A Challenging Tranformation”, In. Proc. of IEEE’s, 2011 Frontiers of Information Technology, Page No- 33-38.ISBN-978-0-7695-4625-4.

[16] Imran S. Bajwa, M. Asif Naeem, Riaz-Ull-Amin, and Dr. M. Abbas Choudhary. 2006, Speech Language Processing Interface for Object-Oriented Application Design using a Rule-Based Framework, In. Proc. Proceedings of 4th International Conference on Computer Applications, Rangoon, Myanmar, Feb 23-24.

[17] Elbendak, Mosa, Paul Vickers, and Nick Rossiter. "Parsed use case descriptions as a basis for object-oriented class model generation." Journal of Systems and Software 84.7 (2011): 1209-1223.

[18] Imran Sarwar Bajwa, Irfan Hyder, “UCD-Generator A LESSA Application for Use Case Design”, In. Proc. IEEE-International Conference on Information and Emerging Technologies, IEEE-ICIET, Pages 200-205 Karachi-Pakistan.

[19] Harmain, H. M., and Robert Gaizauskas. "Cm-builder: A natural language-based case tool for object-oriented analysis." Automated Software Engineering 10.2 (2003): 157-181.

[20] Sagar, Vidhu Bhala, Vidya, and S. Abirami. "Conceptual modeling of natural language functional requirements." Journal of Systems and Software 88 (2014): 25-41.

[21] More, Priyanka, and Rashmi Phalnikar. "Generating UML diagrams from natural language specifications." International Journal of Applied Information Systems 1.8 (2012): 19-23.

[22] Fabian Friedrich, Jan Mendling and Frank Puhlmann. 2012, Process Model Generation from Natural Language Text, In Advanced Information Systems Engineering, Eds. Lecture Notes in
Mich, Luisa, and Roberto Garigliano. "A linguistic approach to the development of object oriented systems using the nl system lolita." *International Symposium on Object-Oriented Methodologies and Systems*. Springer, Berlin, Heidelberg, 1994.

Harmain, Harmain Mohamed, and R. Gaizauskas. "CM-Builder: an automated NL-based CASE tool." *Proceedings ASE 2000. Fifteenth IEEE International Conference on Automated Software Engineering*. IEEE, 2000.

Harmain, H. M., and Robert Gaizauskas. "Cm-builder: A natural language-based case tool for object-oriented analysis." *Automated Software Engineering* 10.2 (2003): 157-181.

Subramaniam, Kalaivani, et al. "UCDA: Use Case Driven Development Assistant Tool for Class Model Generation." *SEKE*. Vol. 7804. 2004.

Clavel, Manuel, Marina Soledad Egea González, and Viviane Torres da Silva. "The MOVA tool: a rewriting-based UML modeling, measuring, and validation tool." *Actas de las XII Jornadas de Ingeniería del Software y Bases de Datos*. 2007.

Ben Abdessalem Karaa, Wahiba, et al. "Automatic builder of class diagram (ABCD): an application of UML generation from functional requirements." *Software: Practice and Experience* 46.11 (2016): 1443-1458.

Naeem M, Bajwa IS, Chaudhri AA, Ali S," A controlled Natural Language Interface to Class Models",13th International Conference on Enterprise Information Systems (ICEIS’11), China, pages 102 – 110, SciTePress, 2011.

Omer Salih Dawood and Abd-El-Kader Sahraoui, "From Requirements Engineering to UML using Natural Language – Survey Study", EJERS, European Journal of Engineering Research and Science, Vol. 2, No. 1, January 2017, 44-50

“About the Unified Modeling Language Specification Version 2.5”, https://www.omg.org/spec/UML/2.5/About-UML/

Hnatkowska, Bogumila, Zbigniew Huzar, and Lech Tuzinkiewicz. "Data modeling with UML 2.0." *Proceedings of the 2005 conference on Software Engineering: Evolution and Emerging Technologies*. 2005.

Nakul Sharma, Prasanth Yalla, “A Hybrid Weighted Probabilistic Based Source Code Graph Clustering Algorithm For Class Diagram And Sequence Diagram Visualization”, International Journal of Scientific & Technology Research Volume 9, Issue 4, Pages 3124-3158.

Mohammadi, Sina, and Habib Izadkhah. "A new algorithm for software clustering considering the knowledge of dependency between artifacts in the source code." *Information and Software Technology* 105 (2019): 252-256.

V. U. Gómez, S. Ducasse and T. D’Hondta, Visually characterizing source code changes, Science of Computer Programming.

S. L. Abebe and P. Tonella, Extraction of domain concepts from the source code, Science ofComputerProgramming98 (2015)680–706.

S. Bajracharya, J. Oshrer and C. Lopes, Sourcerer: An infrastructure for large-scale collection and analysis of open-source code, Science of Computer Programming 79 (2014) 241–259.

Abilhoa, Willyan D., and Leandro N. De Castro. "A keyword extraction method from twitter messages represented as graphs." Applied Mathematics and Computation 240 (2014): 308-325.

Ziqi Zhang, Johann Petruk, Diana Maynard, Adapted TextRank for Term Extraction: A Generic Method of Improving Automatic Term Extraction Algorithms, Procedia Computer Science, Volume 137, 2018, Pages 102-108, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2018.09.010.

Duari, Swagata, and Vasudha Bhatnagar. “Complex Network Based Supervised Keyword Extractor.” Expert Systems with Applications 140 (2020): 112876.

Wael Etaiwi, Arafat Awajan, Graph-based Arabic text semantic representation, Information Processing & Management, Volume 57, Issue 3, 2020, 102183, ISSN 0306-4573, https://doi.org/10.1016/j.ipm.2019.102183.
[42] Javad Rafiei-Asl, Ahmad Nickabadi, TSAKE: A topical and structural automatic keyphrase extractor, Applied Soft Computing, Volume 58, 2017, Pages 620-630, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2017.05.014.

[43] Shansong Yang, Weiming Lu, Dezhi Yang, Xi Li, Chao Wu, Baogang Wei, KeyphraseDS: Automatic generation of survey by exploiting keyphrase information, Neurocomputing, Volume 224, 2017, Pages 58-70, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2016.10.052.

[44] B. D. Farahani, S. O. Fatemi and M. Ghorbani, "Automatic Keyphrase Extraction from Persian Scientific Documents Using Semantic Relations," 2019 27th Iranian Conference on Electrical Engineering (ICEE), Yazd, Iran, 2019, pp. 1972-1978, doi: 10.1109/IranianCEE.2019.8786696.

[45] Kousar Nikhath, A., & Subrahmanyam, K. (2019). Feature selection, optimization and clustering strategies of text documents. International Journal of Electrical and Computer Engineering, 9(2), 1313-1320. doi:10.11591/ijee.v9i2.pp.1313-1320

[46] Niraj Kumar, Dr. Kannan Srinathan, Dr. Vasudeva Varma, Towards Intelligent Text Mining Under Limited Linguistic Resources. PhD Thesis, IIIT Hyderabad, http://hdl.handle.net/10603/120959.

[47] Litvak, M., Last, M. & Kandel, A. DegExt: a language-independent keyphrase extractor. J Ambient Intell Human Comput 4, 377–387 (2013). https://doi.org/10.1007/s12652-012-0109-z

[48] Kumar, Niraj, Kannan Srinathan, and Vasudeva Varma. "A graph-based unsupervised N-gram filtration technique for automatic keyphrase extraction." International Journal of Data Mining, Modeling and Management 8.2 (2016): 124-143. DOI: 10.1504/IJDMMM.2016.077198

[49] Yakoob, Sk. Dr. V. Krishna Reddy and C. Dastagiraiiah, “Analysis of Keyword Searchable Methodologies in Encrypted Cloud Data.” (2016), IJET.

[50] Yan Ying, Tan Qingping, Xie Qinzeng, Zeng Ping, Li Panpan, A Graph-based Approach of Automatic Keyphrase Extraction, Procedia Computer Science, Volume 107, 2017, Pages 248-, ISSN 1877-0509,https://doi.org/10.1016/j.procs.2017.03.087.

[51] S. Beliga, A. Meštrovic, S. Martinic-Ipšic, "An Overview of Graph-Based Keyword Extraction Methods and Approaches", Journal of Information and Organizational Sciences, VOL. 39, NO. 1 (2015), PP. 1-20.

[52] F. Bulgarov, C. Caragea, "A Comparison of Supervised Keyphrase Extraction Models", WWW 2015 Companion, May 18-22, 2015, Florence, Italy, ACM 978-1-4503-3473-0/15/05. http://dx.doi.org/10.1145/2749098.2742776.

[53] K. Saidul Hasan, V. Ng, "Automatic Keyphrase Extraction: A Survey of the State of the Art", Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, Maryland, USA, June 23-25 2014, pages 1262–1273. http://acl2014.org/acl2014/P14-1/pdf/P14-1119.pdf

[54] T. Gupta, "Keyword Extraction: A Review", International Journal of Engineering Applied Sciences and Technology, 2017. Vol. 2, Issue 4, ISSN No. 2455-2143, Pages 215-220.

[55] S. Siddiqi, Aditi Sharan, "Keyword and Keyphrase Extraction Techniques: A Literature Review", International Journal of Computer Applications, Volume 109 – No. 2, January 2015. ISBN: 0975 – 8887.

[56] 56. Jose Mary, G., & Haritha, D. (2017). A survey on best keyword cover search. Journal of Advanced Research in Dynamical and Control Systems, 9(Special issue 14), 2217-2231. Retrieved from www.scopus.com

[57] Mahata, Debajnan, et al. "Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings." Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 2018.

[58] Florescu, Corina, and Cornelia Caragea. "Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents." Proceedings of the 55th Annual Meeting of the
Assoication for Computational Linguistics (Volume 1: Long Papers). 2017.

[59] Josiane Mothe, Faneka Ramiadrisoa, Michael Rasolomahanana. Automatic keyphrase extraction using graph-based methods. SAC 2018: The 33th ACM Symposium on Applied Computing, Apr 2018, Pau, France. pp.728-730. HAL Id: hal-02640988. https://hal.archives-ouvertes.fr/hal-02640988

[60] Degl'Innocenti, Dante, Dario De Nart, and Carlo Tasso. "A New Multi-lingual Knowledge-base Approach to Keyphrase Extraction for the Italian Language." KDIR. 2014.

[61] Ravinuthala, V. V. M. K., and Satyananda Reddy Chinnam. "A keyword extraction approach for single document extractive summarization based on topic centrality." International Journal of Intelligent Engineering Systems (2017).

[62] P. Selvaraj, V. K. Burugari, D. Sumathi, R. K. Nayak and R. Tripathy, "Ontology based Recommendation System for Domain Specific Seekers," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2019, pp. 341-345, doi: 10.1109/I-SMAC47947.2019.9032634.

[63] Biswas, Saroj Kr, Monali Bordoloi, and Jacob Shreya. "A graph based keyword extraction model using collective node weight." Expert Systems with Applications 97 (2018): 51-59.

[64] Vijay Bhaskar Reddy, Y., Reddy, L. S. S., & Sai Satya Naryana Reddy, S. (2017). Comparative study of density-based clustering algorithms. International Journal of Civil Engineering and Technology, 8(12), 763-767. Retrieved from www.scopus.com

[65] Kishore, P. V. V., Anantha Rao, G., Kiran Kumar, E., Teja Kiran Kumar, M., & Anil Kumar, D. (2018). Selfie sign language recognition with convolutional neural networks. International Journal of Intelligent Systems and Applications, 10(10), 63-71. doi:10.5815/ijisa.2018.10.07

[66] Prasada Rao, C., Siva Kumar, P., Rama Sree, S., & Devi, J. (2018). An agile effort estimation based on story points using machine learning techniques doi:10.1007/978-981-10-8228-3_20 Retrieved from www.scopus.com

[67] Ahad, A., Yalavarthi, S. B., & Hussain, M. A. (2018). Tweet data analysis using topical clustering. Journal of Advanced Research in Dynamical and Control Systems, 10(9 Special Issue), 632-636. Retrieved from www.scopus.com

[68] V MNSSVKR Gupta, CH.V. Phani Krishna, “Key Node Selection Network Analysis And Centrality Measurements On A Dataset Of Cancer Documents”, VOL. 14, NO. 5, MARCH 2019. ISSN 1819-6608.

[69] Aziz Nanthamornphong, Anawat Leatongkam, "Extended ForUML for Automatic Generation of UML Sequence Diagrams from Object-Oriented Fortran" Hindawi, Scientific Programming, Volume 2019. https://doi.org/10.1155/2019/2542686