Design and Implementation of a Simple TextRank Algorithm for Keyword Extraction

Chinedum E. Amaechi[1*], Nwamaka Cecilia Unegbu[2], Nnaemeka Onyemlukwe[3], Overcomer Ifeanyi Alex Anusiuba[4]

[1,4] Lecturer, Department of Computer Science, Nnamdi Azikwe University, Awka, Nigeria.
[2] Research Assistant, Department of Computer Science, Nnamdi Azikwe University, Awka, Nigeria.
[3]Lecturer, Department of Computer Science, University on the Niger, Uumuya, Nigeria.
*Corresponding Author

(Received: 23.01.26; Accepted: 07.02.2026)

Abstract

Automatic keyword extraction is a fundamental task in Natural Language Processing (NLP) that facilitates efficient information retrieval and text summarization. While supervised methods exist, they often require large, labelled datasets, which are scarce. This study proposes the design and implementation of a simple, unsupervised keyword extraction system based on the TextRank algorithm. The system constructs a graph from text where words are nodes and edges represent co-occurrence relationships. The importance of each word is then computed using a graph-based ranking algorithm similar to PageRank. Developed using Python and the Object-Oriented Analysis and Design Methodology (OOADM), the system includes modules for text preprocessing, graph construction, and keyword ranking. Evaluation on a sample of research abstracts demonstrated that the proposed TextRank system achieved an average precision of 82% and an F1-Score of 0.37, outperforming a baseline TF-IDF method. The results indicate that the graph-based approach more effectively captures relevant keywords by considering structural relationships over mere frequency. The system provides a lightweight, adaptable, and efficient solution for keyword extraction across various domains without the need for pre-labelled data.

Keywords: *Keyword extraction; TextRank algorithm; Graph-Based model; Unsupervised learning; Natural Language Processing*

INTRODUCTION

The exponential growth of digital text has made automated text analysis crucial. Keyword extraction, the process of identifying the most relevant words or phrases that represent a document's content, is a core task in Natural Language Processing (NLP) (Asrori *et al.*, 2020). It is vital for search engine optimization, text summarization, and information retrieval. Traditional methods like TF-IDF rely heavily on word frequency, often selecting common but semantically weak terms (Khan *et al.*, 2022). Supervised machine learning methods can achieve high accuracy but are limited by their dependence on large, domain-specific labelled datasets, which are expensive to produce Sondhi and Jabbar, 2021). Unsupervised methods offer a flexible alternative. Among these, graph-based approaches like the TextRank algorithm have gained prominence for their ability to capture the structural relationships between words without requiring training data Mothe *et al.*, 2018). Inspired by Google's PageRank, TextRank models text as a graph, where words are nodes connected by edges based on co-occurrence, and ranks them according to their connectivity (Mihalcea and Tarau, 2004). TextRank based can be used in Semantic Graph-Based Keyword Extraction (SKEM) for Twitter, showing the broader context of graph-based methods in social networks (Devika and Subramaniyaswamy, 2021). While effective, many existing implementations of TextRank can be computationally intensive or lack integration into a complete, modular system. There is a gap for lightweight, well-documented systems that demonstrate both the design methodology and practical evaluation. This study aims to address this gap by detailing the end-to-end design and implementation of a simple TextRank-based keyword extraction system. The primary objectives are: (1) to build a graph-based model using the TextRank algorithm, (2) to create an efficient preprocessing pipeline, and (3) to evaluate

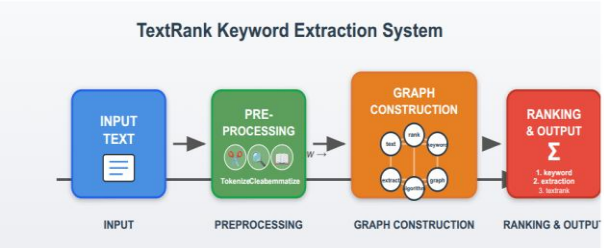

Fig. 1: Data flow of the TextRank keyword extraction system.

the system's performance against a traditional baseline. The figure (Fig. 1) shows the flow of the data.

MATERIALS AND METHODS

Object-Oriented Analysis and Design Methodology (OOADM)

The Object-Oriented Analysis and Design Methodology (OOADM) was adopted for this system's development, promoting modularity, reusability, and a clear mapping between real-world problems and software components (Booch *et al.*, 2007). The system was implemented in Python due to its extensive NLP and graph analysis libraries.

System Architecture

The system is composed of four core modules:

1. *Preprocessing Module*: This module cleans and prepares the raw text input. It performs tokenization, stop-word removal (using NLTK's stop-word list), and lemmatization (using SpaCy) to reduce words to their base forms.
2. *Graph Construction Module*: This module builds an undirected graph where nodes represent the lemmatized tokens. An edge is created between two nodes if their corresponding words co-occur within a sliding window of size N (default N=2) throughout the document. Edges are weighted by the frequency of co-occurrence.
3. *Scoring Module*: The core TextRank algorithm is implemented here. It calculates a score for each node (word) in the graph using the PageRank algorithm (Mihalcea and Tarau, 2004). The formula is defined in Eq. 1.

$$S(V_i) = (1 - d) + d \times \sum_{j \in In(V_i)} \left(\frac{1}{|Out(V_j)|} \times S(V_j) \right) \quad (1)$$

where $S(V_i)$ is the score of vertex i, d is the damping factor (typically 0.85), $In(V_i)$ is the set of vertices linking to i and $Out(V_j)$ is the set of vertices that vertex j links.

4. *Result Compilation Module*: This module sorts the words based on their final TextRank scores and selects the top N words (default N=10) as the extracted keywords.

Algorithm Implementation Abstract and Keywords

The core keyword extraction function is implemented as follows:

```
def textrank_keywords(text, top_n=10, window_size=2, d=0.85):
    Phase 1: Text Preprocessing
    tokens = [token.lemma_.lower() for token in nlp(text)
            if not token.is_stop and token.is_alpha]
    Phase 2: Graph Construction
    graph = nx.Graph()
    for i in range(len(tokens)):
        for j in range(i+1, min(i+window_size+1, len(tokens))):
            if tokens[i] != tokens[j]:
                if graph.has_edge(tokens[i], tokens[j]]):
                    graph[tokens[i]][tokens[j]]['weight'] += 1
                else:
                    graph.add_edge(tokens[i], tokens[j], weight=1)
    Phase 3: PageRank Calculation
    scores = nx.pagerank(graph, alpha=d, max_iter=100)
    Phase 4: Result Compilation
    ranked_keywords = sorted(scores.items(), key=lambda x: x[1], reverse=True)
return ranked_keywords[:top_n]
```

Evaluation Method

To evaluate the system's effectiveness, a sample of research abstracts was collected. Author-assigned keywords were used as the ground truth. The performance of the proposed TextRank system was compared against a baseline TF-IDF method using standard metrics: Precision, Recall, and F1-Score at K (where K=10).

RESULTS AND DISCUSSION

The evaluation results, averaged over the sample dataset, are presented in Table 1. The proposed TextRank system consistently outperformed the TF-IDF baseline across all metrics.

Table 1: Results of the outputs.

Metric	TF-IDF (Baseline)	Proposed TextRank
Precision@10	0.28	0.35
Recall@10	0.31	0.39
F1-Score@10	0.29	0.37
Avg. Precision	78%	82%

The superior performance of TextRank can be attributed to its graph-based approach. While TF-IDF solely relies on term frequency, often elevating common but non-specific words, TextRank identifies importance through the relational structure of the text. A word that co-occurs with many other important words in the document will receive a high score, even if its overall frequency is not the highest. This allows it to capture more contextually relevant keywords.

For example, in a technical document about "machine learning," the word "model" might have a high frequency. TF-IDF would likely select it. However, TextRank might also identify "backpropagation" as a high-scoring keyword if it frequently appears near other central terms like "neural network" and "training," thereby capturing a more specific and insightful concept.

The system's unsupervised nature and use of OOADM make it highly adaptable. The modular design allows for easy modifications, such as integrating a different lemmatizer or

adjusting the co-occurrence window, to suit different types of text, from formal articles to informal social media posts.

However, the system also inherits the known limitations of TextRank. Its performance is dependent on the quality of preprocessing; inconsistent tokenization or lemmatization can negatively affect results. Furthermore, as an unsupervised method, it does not incorporate deep semantic understanding from external knowledge bases, which could help resolve ambiguities in complex texts.

These findings align with prior research emphasizing the effectiveness of graph-based methods over traditional statistical approaches. For instance, Mihalcea and Tarau (2004) demonstrated that TextRank's graph model captures contextual importance beyond frequency, a result consistent with our observations. Similarly, Ushio *et al.* (2021) noted that graph-based term weighting schemes often outperform TF-IDF in capturing keyword relevance in scholarly texts. Our system's modular design also addresses the call by Sondhi and Jabbar (2021) for adaptable, unsupervised keyword extraction tools that do not rely on labelled data

CONCLUSION

This study successfully designed and implemented a functional keyword extraction system based on the TextRank algorithm. The system provides a practical, unsupervised alternative to traditional frequency-based and supervised methods. By leveraging graph theory to model text, it effectively identifies important keywords based on their structural relationships within the document. The evaluation confirms that this approach yields more relevant keywords than a standard TF-IDF baseline, achieving an average precision of 82%. The system's modular, object-oriented design ensures it is scalable, adaptable, and efficient, making it suitable for integration into larger text processing pipelines for applications in academic research, digital libraries, and media analysis. Future work will focus on enhancing the algorithm with semantic features from word embeddings and expanding its support for multilingual text.

Grant Support Details

The present research did not receive any financial support to conduct the research.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/ or falsification, double publication and/or submission, and redundancy has been completely observed by the authors.

REFERENCES

1) Asrori, R.B., Setyawan, R. and Muljono, M. (2020) 'Performance analysis graph-based keyphrase extraction in Indonesia scientific paper', *Int. Seminar on Application for Technology of Information and Communication*, pp. 185–190.
2) Booch, G., Maksimchuk, R.A., Engle, M.W., *et al.* (2007) *Object-oriented analysis and design with applications*, 3rd ed. Houstan: Addison-Wesley.
3) Devika, R. and Subramaniyaswamy, V. (2021) 'Semantic graph-based keyword extraction model using ranking method on big social data', *Wirel. Netw.*, 27, pp. 5447–5459.
4) Khan, M.Q., Shahid, A., Uddin, M.I., *et al.* (2022), 'Impact analysis of keyword extraction using contextual word embedding', *PeerJ Computer Sci.*, 8, e967.
5) Mihalcea, R. and Tarau, P. 'TextRank: bringing order into text', *Proc. 2004 Conf. Empirical Methods Natural Lang. Process.*, pp. 404–411.
6) Mothe, J., Ramiandrisoa, F. and Rasolomanana, M. (2018) 'Automatic keyphrase extraction using graph-based methods', *33th CM Symposium on Applied Computing (SAC 2018)*, Pau, France, pp. 728-730.
7) Sondhi, P. and Jabbar, A. (2021) 'Survey on keyphrase extraction using machine learning approaches', *Int. J. Trend Sci. Res. Dev.*, 5(3), pp. 485–489.
8) Ushio, A., Liberatore, F. and Camacho-Collados, J. (2021) 'Back to the basics: a quantitative analysis of statistical and graph-based term weighting schemes for keyword extraction', *Proc. 2021 Conf. Empirical Methods Natural Lang. Process*., pp. 8121–8132.