New Term Weighting Algorithm for Single Document Summarization

[1]Olanrewaju BABAJIDE, [2]Victoria OYEKUNLE &[3]Akintayo AYOADE
[1]@neweralanre@gmail.com/+234 7032261917,[2]@bola.oyekunle@lcu.edu.ng/+ 234 803 502 8704; [3]@akintayo.ayoade@lcu.edu.ng/ +234 8033497824

[1 2, 3]Lead City University Ibadan, Nigeria

Abstract
Keyword extraction plays a central role in single-document summarization, where the task is to identify the most salient terms that capture the meaning of a text. Existing unsupervised keyword extraction approaches, such as RAKE, TextRank, and YAKE, rely primarily on frequency and statistical co-occurrence. Although effective, they often overlook the structural and semantic contributions of sentences, which are essential for preserving context, especially in long or complex documents. In this work, we propose Sentencer, a novel unsupervised term weighting algorithm that integrates sentence-level features into keyword scoring. Unlike frequency-only approaches, Sentencer leverages contextual relevance, sentence length, intra-sentence probability, and sentence position to refine keyword importance. The algorithm is evaluated against YAKE on three benchmark datasets: SemEval (scientific papers), Inspec (abstracts), and a collection of news reports. Results show that Sentencer performs particularly well on long, complex texts such as scientific papers, where it achieves superior precision and recall compared to YAKE, albeit at the cost of computational efficiency. Furthermore, Sentencer offers a secondary benefit as a diagnostic tool for analyzing sentence behavior and word distribution dynamics within documents. For short scientific abstracts, Sentencer out performed YAKE by 3.5% while for scientific articles, Sentencer out performed YAKE by 1%. However, for short news articles (100 to 400 words), YAKE outperforms Sentencer.

Keywords: Sentencer, Keyword extraction, Sentence-level features, Contextual semantics, Natural language processing.
Word Count: 210

1. Introduction

Keyword extraction refers to the automatic identification of words or short phrases that capture the main ideas of a document. These keywords serve as semantic anchors that enable readers, researchers, and computational systems to quickly understand the core content of a text without reading it in full. Because of this, keyword extraction is essential across many natural language processing (NLP) applications, including information retrieval, document clustering, topic modeling, and, notably, automatic text summarization (Manning et al., 2008; Liu, 2011).

Broadly speaking, keyword extraction techniques fall into two categories: supervised and unsupervised approaches. Supervised methods typically rely on annotated datasets and external linguistic resources (Witten et al., 1999; Zhang et al., 2016). While such methods can achieve high accuracy within specific domains, they suffer from two major limitations. First, they are expensive to train because they require substantial labeled data. Second, they tend to be domain-dependent. These limitations make supervised approaches impractical for many real-world and low-resource environments where annotated corpora are unavailable.

Unsupervised methods, in contrast, avoid the need for labeled data or external knowledge. They typically rely on intrinsic textual properties such as word frequency, co-occurrence patterns etc. (Turney, 2000; Campos et al., 2020). Their main advantage is domain independence. This flexibility has led to widespread adoption in multilingual and low-resource contexts. However, unsupervised techniques still face important challenges. Most treat words as independent units, emphasizing statistical regularities while overlooking the deeper semantic and structural roles that words play within sentences (Nomoto, 2023).

Despite decades of development, the limitations of existing methods remain evident. Frequency-based approaches often overemphasize common but uninformative terms, while graph-based models such as TextRank, though capturing some structural information, lack the semantic sensitivity required in complex or technical domains (Rose et al., 2010). Hybrid systems like YAKE improve robustness by integrating additional statistical cues but still rely primarily on coarse text-wide statistics (Campos et al., 2020). These constraints point to the need for models that treat sentences as active structures that influence term importance.

To address these shortcomings, this study proposes Sentencer, a new unsupervised keyword extraction algorithm that places sentence-level behavior at the center of term weighting. Rather than evaluating words in isolation, Sentencer models how a term behaves within its sentence environment by integrating contextual, probabilistic, and positional features.

The goal of this work is to bridge the gap between frequency-based efficiency and semantically informed extraction. By incorporating sentence-aware features, Sentencer aims to produce more accurate and interpretable keyword selections, particularly in domains where contextual sensitivity is essential, such as scientific or technical writing.

2. Related Work

The motivation for unsupervised methods is rooted in their domain independence and adaptability, making them suitable for applications where labeled data are scarce or unavailable. Early probabilistic approaches to keyword extraction were heavily influenced by research in information retrieval. For example, early methods such as Relevance Weighting used probabilistic term-weighting functions (Sparck Jones, 1972; Mustafi, 2023). This approach was effective for its time but lacked semantic sensitivity. The Okapi BM25 model (Robertson & Walker, 1994) later advanced probabilistic retrieval with a bag-of-words scoring function. However, BM25 and related models assume independence between terms.

As limitations of purely probabilistic models became evident, graph-based approaches emerged. The most notable example is TextRank (Nomoto, 2023). Although TextRank provided a more structural perspective compared to frequency-only models, empirical studies later showed that graph-based methods do not always outperform simpler statistical models for keyword extraction (Nomoto, 2023). Other graph-based innovations include methods that integrate semantic similarity measures (Mihalcea & Csomai, 2007), or those that combine graph ranking with word embeddings to capture latent semantic relationships (Litvak & Last, 2008). While these approaches enrich graph connectivity with semantic cues, they often come at the cost of increased computational complexity and dependence on external lexical resources.

Another widely used family of unsupervised techniques is candidate phrase segmentation methods, among which RAKE (Rapid Automatic Keyword Extraction) stands out (Rose et al., 2010). While RAKE has the advantage of efficiency and simplicity, it can sometimes produce overly long or irrelevant key-phrases (Boumans & Trilling, 2016).

More recently, YAKE (Yet Another Keyword Extractor) was proposed as an improvement over earlier statistical approaches by incorporating multiple local features into its scoring function (Campos et al., 2020). Although YAKE demonstrates competitive performance and remains computationally lightweight, its representation of semantic relationships is relatively coarse.

While each generation of methods has improved upon its predecessors, significant limitations remain. Probabilistic and frequency-driven models tend to neglect contextual meaning, graph-based approaches struggle with efficiency and generalizability, and hybrid models like YAKE provide only partial solutions by combining multiple local features.

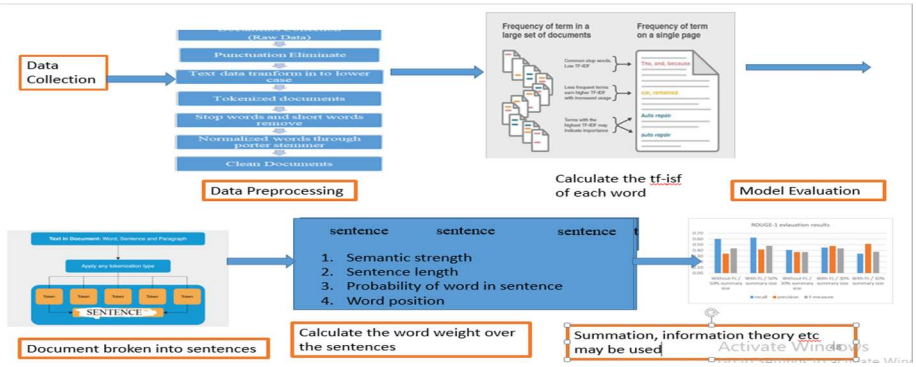
For the supervised models, transformer based models have being used. Peizek et al, (2022) used the T5 transformer model (which is an encoder-decoder model) to perform keyword extraction on short polish text. The result was promising. Also, Bassem et al, (2024) used prompt engineering (which is based on deep learning) for keyword extraction of Arabic text.

This landscape highlights the gap that motivates the present study i.e., the need for an algorithm that not only leverages frequency and distributional statistics but also incorporates sentence-aware features that reflect the structural and semantic roles of words in their immediate contexts. The proposed Sentencer algorithm builds directly upon this gap, aiming to capture sentence-level dynamics in a manner that enhances both interpretability and keyword extraction performance.

3. Methodology

The proposed algorithm, Sentencer, introduces a novel approach to term weighting that incorporates sentence-level behavior of words. Unlike frequency-only methods that assume a word's importance is constant across the entire document, Sentencer models how the significance of a term shifts depending on its local context. This section describes the design of the algorithm, its feature representation, and the aggregation mechanism used to compute final term weights. The framework of the methodology is shown in the figure below.

Figure 1.1



3.1 Feature Representation

The overall features considered by the algorithm is emphasized on below:

Contextual Weight (V1): Context plays a critical role in determining the meaning and importance of a word. Traditional approaches such as term frequency–inverse document frequency (tf–idf) ignore such local variations because they treat term importance as fixed across a document or

corpus (Manning et al., 2008). To address this, Sentencer adopts a refinement of the term frequency–inverse sentence frequency (tf–isf) method, which shifts the unit of measurement from documents to sentences.

Building on this idea, we propose a trigonometric transformation of tf–isf to better capture semantic variation within sentences. The approach adapts the cosine rule from geometry, where the weight of a word is modeled as the length of a triangle side determined by the present tf–isf value, its preceding value, and its deviation from the sentence mean (Hadi et al., 2020). The formulation is expressed as:

$$W(semantic) = \frac{a^2+b^2-c^2}{2ab} \quad (1)$$

Where a represents the present tf-isf value, of the word, b the preceding tf–isf value and c is the deviation of the tf-isf value of the word from the mean tf-isf of the words in the document. By integrating sentence-level variance into the weighting function, this transformation provides a dynamic assessment of importance, ensuring that words appearing in semantically dense contexts are given proportionally higher weight. The use of trigonometric models for term weighting aligns with prior work in NLP. For example, trigonometric relations have been applied to spam classification, where triangle sides represented word frequencies in different classes (Hadi et al, 2022), and positional encoding in transformer models often employs sine and cosine functions (Vaswani et al, 2023). Moreover, Fourier analysis (which is used in textual analysis of data) also relies on trigonometric functions to capture complex patterns (Hussein et al, 2023). These precedents underscore the validity and potential of trigonometric approaches in NLP applications. Sentence Length (V2): Sentence length has long been recognized as a proxy for information richness (Halliday & Hasan, 1989). Longer sentences typically carry more propositional content and encode more relationships between terms (Sarker et al, 2020). To reflect this, Sentencer introduces a normalization factor that scales the weight of words according to the length of the sentence in which they appear. Words in longer, information-dense sentences receive slightly higher scores, balancing against the tendency of frequency-only approaches to overweight short, repeated phrases.

Intra-Sentence Word Probability (V3): Another dimension of word salience is the probability of occurrence within its own sentence. Sentencer computes this as the relative

frequency of a word normalized by sentence length. The intuition is that words that recur within a sentence are central to its meaning. This draws inspiration from information theory, where local entropy and redundancy indicate importance (Shannon, 1948; Anbukkarasi et al., 2022). The metric for measuring the intra-sentence probability is shown below.

$$\text{V3 (inter-sentence probability)} = \frac{tf-isf}{\sum(tf-i\quad)} \qquad (2)$$

Where tf-isf = tf-isf value of the individual word in the sentence and
∑(tf-isf) = summation of the tf-isf values of all the words in the sentence.

Sentence Position (V4): Finally, Sentencer accounts for the position of a sentence within the overall document. Previous studies have shown that sentence position is a strong indicator of salience, particularly in structured genres such as news reports and academic writing (Edmundson, 1969; Lin & Hovy, 1997). Introductory sentences often contain key background information, while concluding sentences frequently summarize or emphasize important findings. To model this, Sentencer applies a positional weight that boosts terms appearing in the first and last few sentences of a document, while discounting those in the middle.
Together, these four features provide a multidimensional representation of word importance.

Feature Aggregation
To compute the final importance of each word, Sentencer aggregates the four feature scores into a single scalar value:

$$W_{\text{final}} = \frac{\sum_{i=1}^{4} V_i}{N} \qquad (3)$$

Where $\sum_{i=1}^{4} V_i$ represents the aggregation of the features which are contextual weight, sentence length normalization, intra- sentence word probability, sentence position.
N stands for the sentences in which the words occur.
Preliminary experiments tested alternative aggregation methods, such as weighted linear combinations, multiplicative scoring etc. However, empirical results suggested that a simple

summation yielded both stable and interpretable scores, outperforming more complex methods in terms of both consistency and ease of analysis.

The Sentencer Algorithm

```
for each sentence i where w occurs:
    v1 = SemanticStrength(w, S[i])
    v2 = IntraSentenceProbability(w, S[i])
    v3 = SentenceLengthFactor(S[i])
    v4 = PositionWeight(i, |S|)
    score += (v1 + v2 + v3 + v4)
TW(w) = score / number_of_occurrences
```

3.2 Experimental Setup

To rigorously evaluate the performance of the proposed Sentencer algorithm, a series of controlled experiments were conducted on three benchmark datasets that vary in length, domain, and complexity.

3.2.1 Datasets

SemEval Dataset: The first dataset comes from the SemEval-2010 Task 5 on automatic key-phrase extraction from scientific articles (Kim et al., 2010). It contains 100 full-length research articles in the computer science domain, with document lengths ranging from 3,000 to 15,000 words. This dataset provides gold- standard human annotations of key-phrases, allowing for direct comparison between algorithmic predictions and expert judgments. Scientific papers were chosen because they represent complex texts with dense information.

Inspec Dataset: The second dataset is the Inspec collection (Hulth, 2003), consisting of 500 scientific abstracts. Each abstract ranges from 100 to 300 words, accompanied by manually assigned keywords. Abstracts are particularly challenging for keyword extraction because they are concise but highly information-dense.

News Dataset: Sourced from the New York Times Annotated Corpus (Sandhaus, 2008), this dataset contains 529 news articles (100–400 words). These articles typically range between 100 and 400 words and are written in a narrative style. Unlike scientific texts, news articles tend to use simpler language, shorter sentences, and more direct structures.

3.2.2 Preprocessing

Before applying the algorithms, all datasets were subjected to a standardized preprocessing pipeline:

Tokenization: Each document was segmented into tokens using a standard word tokenizer, ensuring consistency across datasets (Bird et al., 2009).

Stop-word Removal: Common function words (e.g., the, of, and) were removed using the NLTK English stop-word list. This prevents trivial terms from dominating the weighting scheme.

Lowercasing: All tokens were lowercased to normalize term variations (e.g., Network vs. network).

Sentence Splitting: Documents were segmented into sentences to allow sentence-level feature computation. Sentence boundary detection was performed using rule-based punctuation heuristics combined with NLTK's pre-trained Punkt tokenizer.

Preprocessing was intentionally kept simple and domain-agnostic to preserve the unsupervised nature of the algorithm. No domain-specific thesauri or annotated corpora were used.

3.3 Evaluation Metrics

Algorithmic outputs were evaluated against human-annotated gold standards using standard metrics from information retrieval:

Precision (P): The proportion of extracted keywords that match human-assigned keywords.

Recall (R): The proportion of human-assigned keywords successfully retrieved by the algorithm.

F1-Score (F1): The harmonic mean of precision and recall.

A loose matching criterion was adopted to allow partial matches (e.g., neural network vs. network) to count as correct, consistent with evaluation practices in prior keyword extraction studies (Kim et al., 2010; Campos et al., 2020). This study focuses exclusively on unigram (single-word) keywords. Multi-word expressions are intentionally excluded to narrow the experimental scope and allow precise evaluation of the proposed term-weighting features.

3.4 Baselines and Comparisons

To benchmark performance, Sentencer was compared against YAKE (Campos et al., 2020), a widely used unsupervised keyword extraction algorithm. YAKE was chosen because it represents the current state-of-the-art among statistical unsupervised approaches, incorporates multiple local features beyond simple frequency, making it a stronger baseline than RAKE or TextRank, and it is computationally lightweight. Both algorithms were restricted to unigram extraction (single-word keywords) for fairness, as multiword phrase extraction introduces additional segmentation complexity not central to this study.

3.5 Results and Discussion

The performance of Sentencer was evaluated across three benchmark datasets: SemEval scientific articles, Inspec scientific abstracts, and a collection of news articles against the unsupervised baseline YAKE. Results are reported in terms of precision, recall, and F1-score.

3.5.1 Scientific Articles (SemEval Dataset)

On the SemEval dataset of long scientific articles (Table 1), Sentencer achieved an F1-score of 50.3, compared to YAKE's 49.3. Both precision and recall were modestly higher for Sentencer (45.1 vs. 44.2 for precision; 57.4 vs. 56.0 for recall). Although the numerical gains may appear small, they are meaningful in the context of keyword extraction, where differences of 1 to 2% often correspond to noticeable improvements in semantic coverage (Kim et al., 2010).

Table 1: Comparative performance on SemEval dataset

Algorithm	F1	Precision	Recall
Sentencer	50.3	45.1	57.4
YAKE	49.3	44.2	56.0

The superior performance of Sentencer on this dataset highlights its strength in handling long, information-dense documents. Frequency-based models often struggle in such contexts (Sarker et al., 2020). The trade-off is computational cost; Sentencer requires substantially more processing time due to sentence-level analysis.

3.5.2 Scientific Abstracts (Inspec Dataset)

On the Inspec dataset of shorter scientific abstracts (Table 2), Sentencer again outperformed YAKE, with an F1-score of 45.1 versus 41.6. Precision (42.8 vs. 39.3) and recall (47.9 vs. 44.4) both showed improvements. These gains, while moderate, suggest that Sentencer is well- suited not only for long-form documents but also for shorter texts that remain semantically dense. Also, the statistical significance testing (Wilcoxon signed-rank test) carried out to compare the f1 scores of both algoritms on the same data returned a value of 0.0007. This shows a very significant difference between the two algorithms for the dataset ($p < 0.001$).

Table 2: Comparative Performance on Inspec Dataset

Algorithm	F1	Precision	Recall
Sentencer	45.1	42.8	47.9
YAKE	41.6	39.3	44.4

3.5.3 News Articles

On the dataset of short news articles, results diverged (Table 3). Here, YAKE achieved an F1-score of 39.6, outperforming Sentencer's 34.4. Both precision (39.3 vs. 34.1) and recall (40.0 vs. 34.8) favored YAKE. Also, a Wilcoxon signed-rank test on paired per-document F1 scores of 0.00007 indicates a statistically significant difference between the two algorithms ($p < 0.001$)

Table 3: Comparative performance on News dataset

Algorithm	F1	Precision	Recall
Sentencer	34.4	34.1	34.8
YAKE	39.6	39.3	40.0

This outcome suggests that sentence-level features add limited value in short, narrative-style texts. News articles often emphasize redundancy, key terms are repeated multiple times in different sentences, making frequency-based baselines highly effective (Rose et al., 2010).

Across all three datasets, performance differences between Sentencer and YAKE were modest, typically within 1 to 5% in F1-score. However, these differences reveal important insights:

Document Complexity Matters: Sentencer excels on long, complex, or technical texts (e.g., scientific papers and abstracts), but underperforms on shorter, less complex texts (e.g., news articles).

Computational Cost vs. Accuracy: Sentencer trades efficiency for interpretability and semantic precision. While YAKE remains more practical for large-scale applications, Sentencer may be preferable in specialized contexts where accuracy is critical.

Feature Interpretability: Unlike embedding-based models (e.g., BERT; Devlin et al., 2019), Sentencer provides transparent feature contributions. This makes it valuable as a diagnostic tool for analyzing how sentence position, length, and local probability influence keyword salience.

Beyond keyword extraction, Sentencer opens avenues for studying the interaction between word frequency and sentence behavior. Such analyses could support a variety of applications, including: authorship attribution, genre classification and discourse analysis.

4. Conclusion

This study introduced Sentencer, a novel unsupervised algorithm for keyword extraction designed to incorporate sentence-level features into the term-weighting process. Unlike conventional approaches such as RAKE, TextRank, or YAKE, which primarily rely on frequency counts, co- occurrence patterns, or distributional heuristics, Sentencer integrates contextual relevance, sentence length normalization, intra-sentence word probability, and sentence position into a unified framework. By combining these dimensions, Sentencer aims to capture not just how often a word occurs but how it functions within its immediate sentence context.

Experimental results across three benchmark datasets, SemEval scientific papers, Inspec abstracts, and New York Times news articles, demonstrated that Sentencer provides consistent improvements in precision and recall on long or technical texts. The gains were most notable in scientific articles and abstracts. On shorter and simpler texts, however, YAKE was better.

A key limitation identified in this study is computational efficiency. Sentencer was significantly slower than YAKE, with runtime differences on the order of hundreds of times per document. Another limitation lies in its reliance on relatively simple sentence-aware heuristics; although effective, these features may not fully capture deeper semantic phenomena.

Despite these challenges, the findings suggest that Sentencer contributes both practical utility and theoretical insight. Practically, it offers a more semantically sensitive framework for keyword extraction, especially valuable in domains such as scientific publishing. Theoretically, it highlights the importance of sentence-level dynamics in shaping word salience, offering a basis for further exploration of how linguistic structure influences keyword extraction.

Looking ahead, several promising avenues for future research can be identified. One important direction concerns optimization and scalability. A second avenue lies in exploring multilingual and low-resource settings. Testing its performance across diverse multilingual corpora, particularly in low- resource contexts, would provide valuable insights.

A third direction involves integration with deep learning methods. Hybrid models that combine Sentencer's interpretable sentence-level features with contextual embeddings from transformer- based architectures, such as BERT or XLM-R, may offer a balance between semantic depth and transparency. This type of integration could lead to stronger performance while retaining the interpretability advantage of sentence-aware features.

Another research path involves extending Sentencer to multiword keyword extraction.

Finally, the applications of Sentencer beyond keyword extraction should be considered. The algorithm's sentence-aware weighting scheme offers potential benefits for related tasks such as authorship attribution, discourse analysis, and genre classification.

References

Abeed Sarker, Yuan-Chi Yang, Mohammed Ali Al-Garadi and Aamir Abbas. (2020). A Light-Weight Text Summarization System for Fast Access to Medical Evidence. Frontiers in Digital Health | www.frontiersin.org | Volume 2 | Article 585559.

Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin (2023). Attention Is All You Need. arXiv:1706.03762v7 [cs.CL]

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.

Boumans, J. W., & Trilling, D. (2016). Taking stock of the toolkit: An overview of relevantautomated content analysis approaches and techniques for digital journalism scholars. N Digital Journalism, 4(1), 8–23. https://doi.org/10.1080/21670811.2015.1096598

Bsir Bassem and Mounir Zrigui (2024). Deep learning based transformers for Keyword extraction. https://www.researchgate.net/publication/378923631_Deep_learning_based_transformers_for_Keyword_extraction

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A. M., Nunes, C., Jatowt, A., & Silveira, M. (2020). YAKE! Keyword extraction from single documents using multiple local features. Information Sciences, 509, 257–289. https://doi.org/10.1016/j.ins.2019.09.013

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of NAACL-HLT (pp. 4171–4186). Association for Computational Linguistics.

Edmundson, H. P. (1969). New methods in automatic extracting. Journal of the ACM, 16(2), 264–285.

Hadeel H. Alfartosy, & Hussein K. Khafaji (2023). New Feature Extraction, Reduction, and Classification Method for Documents Based on Fourier Transformation. International Journal of Intelligent Engineering and Systems, Vol.16, No.5

Halliday, M. A. K., & Hasan, R. (1989). Language, Context, and Text: Aspects of Language in a Social-Semiotic Perspective. Oxford University Press.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 216–223). Association for Computational Linguistics.

Kim, S. N., Medelyan, O., Kan, M. Y., & Baldwin, T. (2010). SemEval-2010 Task 5: Automatic keyphrase extraction from scientific articles. In Proceedings of the 5th International Workshop on Semantic Evaluation (pp. 21–26). Association for Computational Linguistics.

Lida Aleksanyan and Armen Allahverdyan. (2024). Unsupervised extraction of local and global keywords from a single text. arXiv:2307.14005v2 [cs.CL] 14 Jun 2024

Lin, C. Y., & Hovy, E. (1997). Identifying topics by position. In Proceedings of the Fifth Conference on Applied Natural Language Processing (pp. 283–290). Association for Computational Linguistics.

Litvak, M., & Last, M. (2008). Graph-based keyword extraction for single-document summarization. In Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization (pp. 17–24). Association for Computational Linguistics.

Liu, B. (2011). Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

Mihalcea, R., & Csomai, A. (2007). Wikify!: Linking documents to encyclopedic knowledge. In Proceedings of the 16th ACM Conference on Information and Knowledge Management (pp. 233–242). ACM.

Piotr Pęzik, Agnieszka Mikołajczyk, Adam Wawrzyński, Bartłomiej Nitoń, Maciej Ogrodniczuk (2022). Keyword Extraction from Short Texts with a Text-To-Text Transfer Transformer. arXiv:2209.14008

Rathi R. N., A. Mustafi. (2023). The importance of Term Weighting in semantic understanding of text: A review of techniques. Multimedia Tools and Applications 82:9761–9783, https://doi.org/10.1007/s11042-022-12538-3

Robertson, S. E., & Walker, S. (1994). Some simple effective approximations to the 2–Poisson model for probabilistic weighted retrieval. In Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 232–241). Springer. https://doi.org/10.1007/978-1-4471-2099-5_24

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. In M. W. Berry & J. Kogan (Eds.), Text Mining: Applications and Theory (pp. 1–20). Wiley

Sandhaus, E. (2008). The New York Times Annotated Corpus. Linguistic Data Consortium, Philadelphia. https://catalog.ldc.upenn.edu/LDC2008T19

Shannon, C. E. (1948). A mathematical theory of communication. Bell System Technical Journal, 27(3), 379–423.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, 28(1), 11–21.

Suha Mohammed Hadi, Ali Hakem Alsaeedi, Dhiah Al-Shammary, Zaid Abdi Alkareem Alyasseri, Mazin Abed Mohammed, Karrar Hameed Abdulkareem, Riyadh Rahef Nuiaa, Mustafa Musa Jaber. (2022). Trigonometric words ranking model for spam message classification. DOI: 10.1049/ntw2.12063.

Tadashi Nomoto (2023). Keyword Extraction: A Modern Perspective. SN Computer Science 4:92 https://doi.org/10.1007/s42979-022-01481-7

Turney, P. D. (2000). Learning algorithms for keyphrase extraction. Information Retrieval, 2(4), 303–336.

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (1999). KEA: Practical automatic keyphrase extraction. In Proceedings of the Fourth ACM Conference on Digital Libraries (pp. 254–255). ACM.

Zhang, J., Zhao, Y., & LeCun, Y. (2016). Character-level convolutional networks for text classification. In Advances in Neural Information Processing Systems (pp. 649–657).