WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Keyword Extraction Based on Semantic Similarity using Wildcard

K. Priyadharshini[1]; Dr.V. Vaidehi[2]

PG Student[1]; Professor[2]
Department of Computer Applications, DR MGR Educational and Research institute Chennai-600095
priyakabaliswaran@gmail.com[1]; vaidehi.mca@drmgrdu.ac.in[2]

ABSTRACT—

Extracting keywords is an effective approach for distilling the main content of a text. However, swiftly extracting useful information from numerous documents poses a challenge. This paper introduces a novel algorithm for keyword extraction, leveraging semantic similarity metrics and multi-feature computation. Initially, a semantic similarity algorithm, SSDIPA (Semantic Similarity based on Distance, Information, and Property Analysis), is proposed to gauge word associations. Subsequently, a multi-feature tuple approach incorporating word frequency, length, span, position, and semantic similarity is introduced to assess candidate keywords. Finally, a feature decision tree adjusts the proportional relationships among these features to align with user preferences. Results indicate that the proposed algorithm surpasses other comparative methods, highlighting the efficacy of the multi-feature computation-based keyword extraction algorithm in enhancing accuracy.

Keywords— **Keywords extraction, Semantic information, Multi feature computing, Subjective preference**

Introduction

As the Internet becomes increasingly pervasive and network technology advances rapidly, the volume of available documents has surged. Extracting pertinent information from this vast corpus swiftly presents a formidable challenge. Consequently, there is a growing interest in academia and IT sectors in algorithms and systems for automatically distilling relevant data from extensive document collections. Consequently, keyword extraction technology within Natural Language Processing (NLP) has emerged as a focal point of research. Keyword extraction (KE) refers to the automated process of identifying terms that encapsulate the essence of a document. Generally, KE methods fall into two main categories: supervised and unsupervised learning. Supervised learning involves training a classifier to extract target keywords from new documents, necessitating annotated training data biased towards specific domains. On the other hand, unsupervised methods assign weights to candidate keywords based on evaluation criteria. To overcome the limitations of supervised approaches, there is a pressing need to explore unsupervised keyword extraction methods based on multi-feature computation. This approach has demonstrated effectiveness and high performance, serving as a foundational component for various downstream tasks such as text classification, summarization, and information retrieval.

RELATED WORK

Numerous publications have covered various aspects of keyword extraction [5]. This section focuses on the related work concerning the acquisition of candidate keyword sets and methods for scoring words. In most keyword extraction methods, the initial step involves obtaining the candidate keyword set from the text. Marujo et al. [6] propose a keyword extraction algorithm based on Brown clustering to address lexical variant issues. However, this method overlooks that combined words within clusters may also serve as keywords. [7] compile a list of stop words using XPO6 and Ranks NL databases. Despite its utility, this approach yields unnecessary data, necessitating filtering of the stop words list before application. Liu et al. [8] utilize regular expressions to match noun phrases with specific patterns to form the candidate keyword set but face limitations in recognizing combined words. When using supervised approaches, the process of keyword extraction is approached as a binary classification issue. In order to assess whether or not candidate keywords are, in fact, keywords, classification classifiers are trained. Turney [9] employs the C4.5 decision tree induction algorithm, considering word frequency and position as crucial features for classification. Susan et al. [10] utilize maximum entropy partitioning to classify candidate keywords based on word frequency per class. Nevertheless, variations in information entropy definitions across methods may affect the efficacy of keyword extraction models. Aman et al. [11] construct sentence parse trees and leverage machine learning techniques to extract features from unstructured documents. However, supervised methods necessitate manually annotated training sets, posing challenges in consistency and scalability across domains. Conversely, unsupervised methods do not require model training, rendering them domain-independent and appealing to researchers. These methods primarily encompass statistical, linguistic, and graph-based approaches. Repar et al. [13] propose a graph-based method that consolidates redundant information into meta vertices. However, different parameter settings may yield disparate extraction results. Linguistic methods, like those of Pudota et al. [4],

incorporate linguistic knowledge and n-gram statistics to define feature sets for phrase extraction. Nonetheless, such methods may overlook semantic word relationships. Typical semantic similarity approaches often consider only the shortest semantic distance or information content within a semantic network [15][16]. Zhang et al. [17] introduce a keyword extraction method that combines Word2Vec with Text Rank to extract keywords from documents. However, utilizing Word2Vec for semantic information from external documents may lead to extraction inaccuracies. In contrast to previous works, a novel keyword extraction algorithm is proposed in this study. It introduces a comprehensive semantic similarity approach and extracts multiple features to score candidate keywords. A unique method is presented for obtaining candidate keyword sets, involving word root restoration, stop words list generation, and combined word construction based on relative position and part-of-speech. Moreover, entropy-related methods are not directly used for training the keyword extraction model. Instead, a specialized decision tree method, combining information entropy and decision tree induction algorithms, is devised to fine-tune system parameters.

Extraction of keywords using wildcards and semantic Similarity

n this section, the proposed algorithm for extracting keywords from text is described in detail. The algorithm consists of three discrete phases: in the first, a collection of candidate keywords is obtained; in the second, numerous features of the candidate keywords are extracted and scored; and lastly, the keyword set is produced. The architecture of the proposed algorithm is shown in Fig. 1.

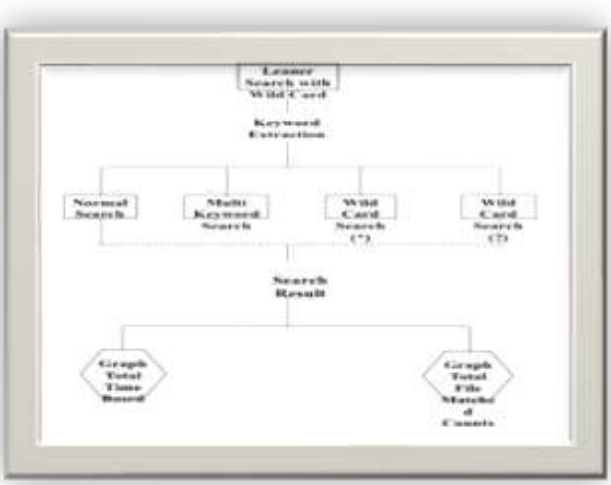

Fig. 1 architecture Diagram

USE CASE DIAGRAM

The Unified Modeling Language (Uml) Plays A Crucial Role in The Creation of Object-Oriented Software and The Whole Software Development Process. The Uml Primarily Use Graphical Notations to Articulate the Software Design of Projects.

A Use Case Diagram in The Unified Modelling Language (Uml) Is A Behavioural Diagram That Is Derived from And Based on A Use-Case Analysis.

The Objective of a Use Case Diagram Is to Visually Depict the Functioning of A System By Illustrating The Actors Involved, Their Goals (Expressed As Use Cases), And Any Interdependencies Between These Use Cases.

The Primary Objective of a Use Case Diagram Is to Illustrate the System Functions That Are Executed for Each Actor. The Roles of The Actors in The System May Be Show.

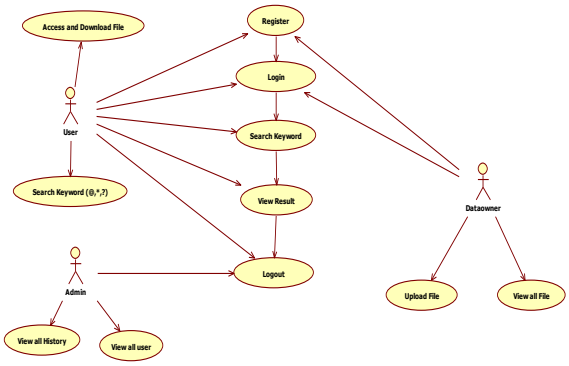

Fig. 2 Use Case Diagram

CLASS DIAGRAM

A class diagram in software engineering, as part of the Unified Modelling Language (UML), is a static structural diagram that provides a description of a system's structure. It illustrates the classes, attributes, operations (or methods), and interactions between the classes. This document provides an explanation of the specific class that includes the relevant information

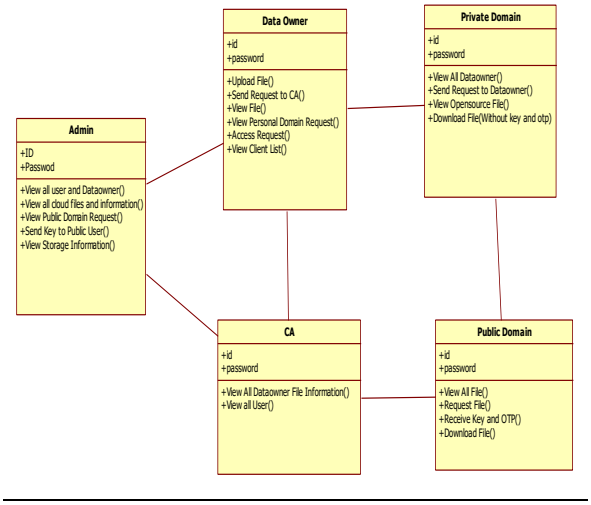

Fig. 3 Class Diagram

SEQUENCE DIAGRAM

A UML sequence diagram is an interaction diagram that illustrates the order and interactions between operations. This is a representation of a Message Sequence Chart. Sequence diagrams are often referred to as event diagrams, event situations, and timing diagrams.

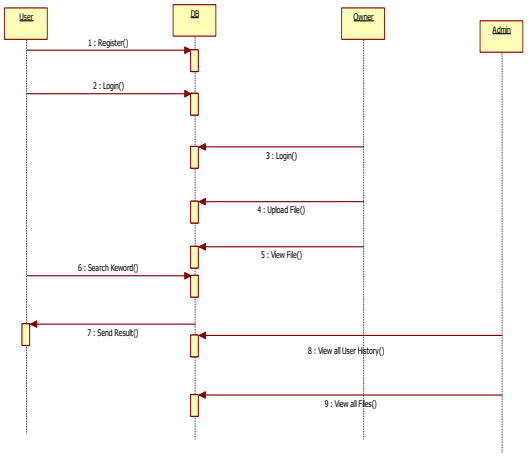

Fig. 4 Sequence Diagram

COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). These diagrams can be used to portray the dynamic behaviour of a particular use case and define the role of each object

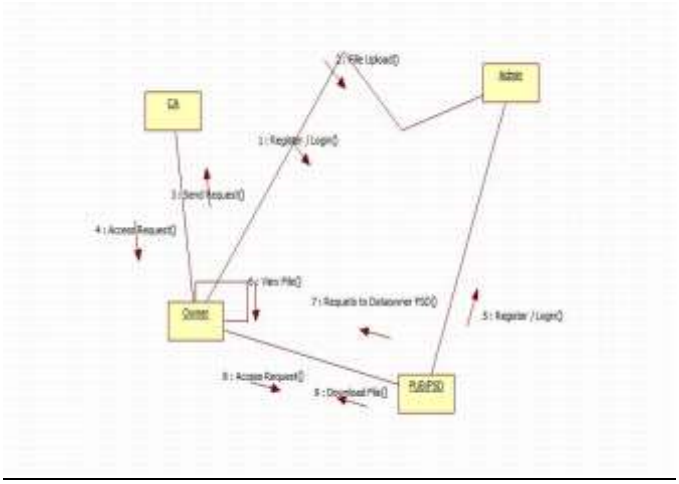

Fig. 5 Collaboration Diagram

DEPLOYMENT DIAGRAM

Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components.

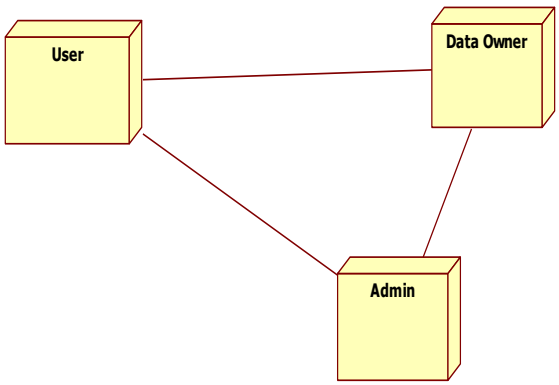

Fig. 5 Deployment Diagram

Conclusions

In the course of this research paper, we have investigated a range of machine learning algorithms that are well-suited for the organization and analysis of enormous quantities of Twitter data. This data comprises millions of tweets and daily text messages that are exchanged. The above-mentioned algorithms, such as the SPC algorithm and linear algebraic factor model approaches, demonstrate remarkable efficacy in handling sizable datasets and facilitate the classification of data into significant categories. It is crucial to acknowledge that the outcomes may differ marginally when the program is executed multiple times, owing to the variability in the retrieved tweets. To address this issue, the program was executed on three separate occasions; the outcomes presented herein are the mean of the three consecutives. Outputs.

The primary objectives of this Endeavor are the development and evaluation of three distinct search mechanisms: a fuzzy search system, a multi-keyword dictionary with Boolean search, and a wildcard keyword search. In addition to enabling queries via AND or OR relationships among multiple keywords, the system integrates the top-k search preferences determined by ranking. .

References

1. Y. Jin, C. Luo, W. Guo, J. Xie, D. Wu, and R. Wang, “Text classification based on conditional reflection,” IEEE Access, vol. 7, pp. 76 712–76 719, 2019
2. A. Hern ´ andez-Casta ´ neda, R. A. Garc ˜ ´ıa-Hernandez, Y. Ledeneva, and ´ C. E. Millan-Hern ´ andez, “Extractive automatic text summarization based ´ on lexical-semantic keywords,” IEEE Access, vol. 8, pp. 49 896–49 907, 2020.
3. J. Zheng and H. Yu, “Key concept identification for medical information retrieval,” in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 579–584.

4. N. Pudota, A. Dattolo, A. Baruzzo, and C. Tasso, “A new domain independent key phrase extraction system,” in Italian Research Conference on Digital Libraries, 2010, pp. 67–78.
5. S. K. Bharti and K. S. Babu, “Automatic keyword extraction for text summarization: A survey,” arXiv preprint arXiv:1704.03242, 2017.
6. L. Marujo, W. Ling, I. Trancoso, C. Dyer, A. W. Black, A. Gershman, D. M. de Matos, J. P. Neto, and J. G. Carbonell, “Automatic keyword extraction on twitter,” in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), 2015, pp. 637–643.
7. M. Zhang, X. Li, S. Yue, and L. Yang, “An empirical study of text rank for keyword extraction,” IEEE Access, vol. 8, pp. 178 849–178 858, 2020.
8. Z. Liu, P. Li, Y. Zheng, and M. Sun, “Clustering to find exemplar terms for key phrase extraction,” in Proceedings of the 2009 conference on empirical methods in natural language processing, 2009, pp. 257–266.
9. P. D. Turney, “Learning to extract key phrases from text,” arXiv preprint cs/0212013, 2002
10. S. Susan and J. Keshari, “Finding significant keywords for document databases by two-phase maximum entropy partitioning,” Pattern Recognition Letters, vol. 125, pp. 195–205, 2019.
11. M. Aman, A. bin Md Said, S. J. A. Kadir, and I. Ullah, “Key concept identification: A sentence parse tree-based technique for candidate feature extraction from unstructured texts,” IEEE Access, vol. 6, pp. 60 403–60 413, 2018.
12. F. Sebastiani, “Machine learning in automated text categorization,” ACM computing surveys (CSUR), vol. 34, no. 1, pp. 1–47, 2002.
13. Skrlj, A. Repar, and S. Pollak, “Rakun: Rank-based keyword ex- ˇ traction via unsupervised learning and meta vertex aggregation,” in International Conference on Statistical Language and Speech Processing. Springer, 2019, pp. 311–323.
14. Z. Zhang, J. Petrak, and D. Maynard, “Adapted text rank for term extraction: A generic method of improving automatic term extraction algorithms,” Procedia Computer Science, vol. 137, pp. 102–108, 2018.
15. Leacock and M. Chodorow, “C-rater: Automated scoring of shortanswer questions,” Computers and the Humanities, vol. 37, no. 4, pp. 389–405, 2003.
16. Lin et al., “An information-theoretic definition of similarity.” in Icml, vol. 98, no. 1998, 1998, pp. 296–304.
17. Y. Zhang, F. Chen, W. Zhang, H. Zuo, and F. Yu, “Keywords extraction based on word2vec and text rank,” in Proceedings of the 2020 The 3rd International Conference on Big Data and Education, 2020, pp. 37–42.
18. S. Rose, D. Engel, N. Cramer, and W. Cowley, “Automatic keyword extraction from individual documents,” Text Mining: Applications and Theory, pp. 1–20, 2010.
19. G. A. Miller, “Wordnet: a lexical database for English,” Communications of the ACM, vol. 38, no. 11, pp. 39–41, 1995.
20. X. Zhu, F. Li, H. Chen, and Q. Peng, “An efficient path computing model for measuring semantic similarity using edge and density,” Knowledge and Information Systems, vol. 55, no. 1, pp. 79–111, 2018.
21. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt, “Yake! collection-independent automatic keyword extractor,” in European Conference on Information Retrieval, 2018, pp. 806–810.
22. Song, Z. Wang, S. Xu, S. Ni, and J. Xiao, “A novel text classification approach based on word2vec and text rank keyword extraction,” in 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC), 2019, pp. 536–543.