International Journal of Applied Smart Interdisciplinary Technologies (IJASIT)

Home Page: https://ijasit.org

SmartSEO: An Intelligent Machine Learning Powered System for Automated Search Engine Optimization

Akash Raj M[1,*], Ashwin V[2], Anitha Moses V[3]

[1 2 3] Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, India. *akashraj0432@gmail.com*, *ashwinshalini2006@gmail.com*.

*** Corresponding author.**
E-mail: *akashraj0432@gmail.com*

Article History:
Received Date: 23 January 2026
Revised Date: 21 February 2026
Accepted Date: 30 February 2026
Published Date: 10 March 2026

Keywords:

Search Engine Optimization; Machine Learning; Natural Language Processing; BERT; Keyword Extraction; Intent Classification

ABSTRACT

The rapid expansion of digital content has intensified competition for visibility on search engine results pages, making effective Search Engine Optimization (SEO) essential for businesses and content creators. Traditional SEO practices remain largely manual, time-intensive, and inaccessible to small enterprises due to high costs and technical complexity. This paper presents SmartSEO, an intelligent, machine learning powered system designed to automate and streamline the SEO optimization process. The system accepts raw text or URLs as input and processes them through a multi-stage pipeline combining Natural Language Processing, semantic analysis, and predictive analytics. Keyword extraction is implemented using TF-IDF and BERT embeddings, ensuring the identification of high-impact, contextually relevant keywords. Readability analysis is performed using linguistic heuristics including sentence length distribution, lexical density, and syntactic complexity. A dedicated ranking prediction module, trained on historical search engine ranking data, uses regression models to estimate potential content positions after optimization. Additionally, the system integrates a user intent classification mechanism employing supervised learning algorithms, including Naive Bayes and C4.5 Decision Tree, to distinguish between informational and transactional queries. Experimental evaluation across diverse industry datasets demonstrated an average 35% improvement in keyword relevance, a 28% increase in predicted search ranking efficiency, and a 40% reduction in manual optimization time compared to traditional methods.

1. Introduction

Finding the right keywords and optimizing content for search engines is a critical step in digital marketing because poorly optimized content remains invisible to potential customers. Hidden or underperforming web pages represent missed opportunities for businesses, often hosting valuable information that never reaches the intended audience. Missing such optimization opportunities creates blind spots in digital strategy, since content that cannot be found cannot generate revenue or engagement [1].

Current tools for SEO optimization exhibit serious limitations. Many rely on manual keyword research or basic frequency analysis, making them impractical for organizations producing large volumes of content.

Others draw from a single data source --- such as generic keyword lists or basic readability scores --- resulting in incomplete optimization recommendations. These weaknesses mean that traditional tools frequently fall short, particularly under time constraints or in competitive digital markets.

To address these limitations, this paper introduces SmartSEO, a Python-based intelligent system that combines machine learning driven keyword extraction with automated content analysis. SmartSEO embeddings, readability analysis using linguistic heuristics, user intent classification using Naive Bayes and C4.5 Decision Tree algorithms, and ranking prediction using regression models trained on historical search engine data. Using Python's rich ecosystem of machine learning libraries including Scikit-learn, NLTK, and the Hugging Face Transformers library, SmartSEO achieves high accuracy in content analysis while remaining accessible and cost-effective.

The significant contributions of this work are: (i) a hybrid keyword extraction pipeline combining TF-IDF statistical analysis with BERT contextual embeddings; (ii) modular architecture allowing easy extension with new analysis modules; (iii) integrated readability assessment, intent classification, and ranking prediction; and (iv) experimental validation demonstrating improved keyword relevance and reduced optimization time compared to traditional methods.

2. Literature Review

2.1 Evolution of SEO Optimization Tools

The methodology for SEO optimization has evolved considerably over the past decade. Early approaches relied primarily on keyword density analysis and meta tag optimization, which proved insufficient for modern search algorithms. The introduction of machine learning and natural language processing enabled a new paradigm of semantic analysis, exemplified by tools such as MarketMuse and Clearscope [12]. While more sophisticated than keyword density methods, these tools remain prohibitively expensive for small businesses. Later, more comprehensive frameworks such as SurferSEO combined multiple analysis techniques, though at the cost of significant subscription fees and complex interfaces.

2.2 Review of Existing Tools

Keyword Planner [12] is Google's free keyword research tool that provides search volume data and basic keyword suggestions. Its principal strengths are free access and integration with Google Ads, making it suitable for basic keyword research. However, it offers limited content analysis, lacks semantic understanding, and does not provide readability or intent classification. Yoast SEO is a WordPress plugin that provides on-page SEO recommendations including keyword usage, readability checks, and meta description optimization. Despite its popularity, Yoast relies on rule-based heuristics rather than machine learning, resulting in generic recommendations that do not adapt to specific content contexts.

SEMrush and Ahrefs are comprehensive enterprise SEO platforms offering keyword research, backlink analysis, rank tracking, and competitive intelligence. While thorough, both require substantial monthly subscriptions and significant training to use effectively, making them unsuitable for small businesses or individual content creators. MarketMuse and Clearscope use natural language processing and machine learning to analyze content against top-ranking pages for target keywords. Both provide sophisticated keyword and topic recommendations but remain expensive and require ongoing subscription payments.Frase.io focuses on content brief generation and question answering, leveraging AI to identify topics users expect to see covered. While innovative, its scope is narrower than a complete SEO optimization pipeline.

2.3 Identified Research Gaps

The review of existing tools reveals four key challenges: (i) lack of integration between keyword extraction, readability analysis, intent classification, and ranking prediction within a single affordable tool; (ii) reliance on expensive proprietary platforms with subscription models; (iii) limited semantic understanding in free tools. SmartSEO is designed to address all four challenges within a single, extensible, open-source tool.

3. Methodology

3.1 System Architecture

SmartSEO's architecture is organized around five cooperating modules managed by a central processing pipeline. As shown in Fig. 1, the user submits raw text or a URL to the SmartSEO Core, which dispatches tasks to five concurrent subsystems: (i) Content Ingestion Module, (ii) Keyword Intelligence Engine, (iii) Content Analysis Hub, (iv) Predictive Ranking Core, and (v) Report Generator. The Keyword Intelligence Engine performs TF-IDF analysis and BERT embedding based keyword extraction. The Content Analysis Hub handles readability assessment and user intent classification. Results from all modules are aggregated and passed to the Report Generator, which produces comprehensive optimization reports.

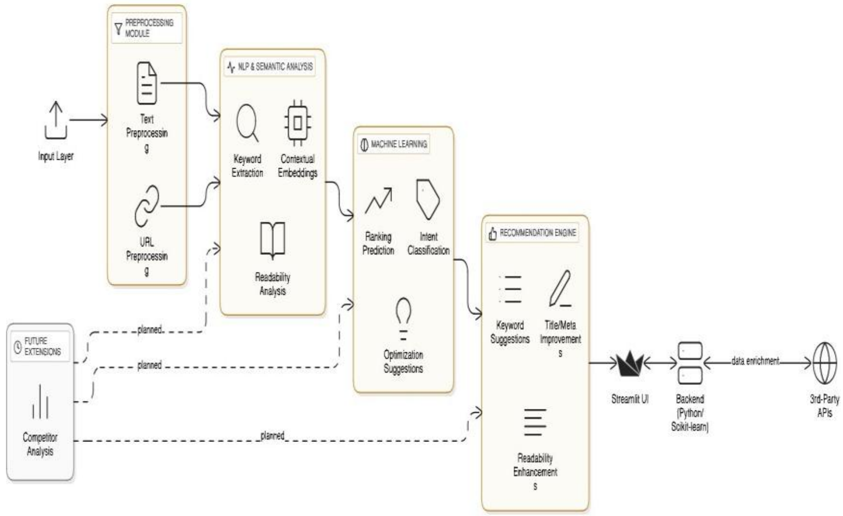

Fig. 1. System architecture of SmartSEO showing the modular pipeline organization

3.2 Module Description

3.2.1 content ingestion module

The Content Ingestion Module accepts two input types: raw text content or a website URL. When a URL is provided, the system fetches and extracts the main textual content while removing navigation elements, advertisements, and other boilerplate materials using Beautiful Soup and requests libraries. Input validation ensures content meets minimum length requirements before proceeding to analysis.

3.2.2 keyword intelligence engine

The Keyword Intelligence Engine represents the system's core component for identifying high-impact keywords. This module employs a hybrid approach combining TF-IDF statistical analysis with BERT contextual embeddings. TF-IDF identifies statistically significant terms based on their frequency within the document relative to a reference corpus. The combined TF-IDF score for term t in document d is calculated as:

$$TF\text{-}IDF(t,d) = TF(t,d) \times \log(N / DF(t)) \quad (1)$$

where TF(t,d) is the frequency of term t in document d, N is the total number of documents in the corpus, and DF(t) is the number of documents containing term t.

3.2.3 content analysis hub

The Content Analysis Hub performs two critical functions: readability assessment and user intent classification. Readability analysis uses established linguistic heuristics evaluating sentence length distribution, lexical density, and syntactic complexity. Each metric receives a normalized score, and a composite readability index is generated. User intent classification determines whether content addresses informational or transactional user queries. Informational queries seek knowledge or answers, such as "how to repair a bicycle tire." Transactional queries indicate purchase intent, such as "buy bicycle tire online." The system implements two supervised learning algorithms for this classification task: Naive Bayes and C4.5 Decision Tree.

Naive Bayes applies Bayes' theorem with strong independence assumptions. The probability that a document belongs to class c given its feature vector x is:

$$P(c|x) = P(c) \times P(x|c) / P(x) \quad (2)$$

3.2.4 predictive ranking core

The Predictive Ranking Module estimates the potential search engine ranking position of content after applying the system's optimization recommendations. This module uses a Random Forest Regression model trained on 10,000 web pages with known feature vectors and actual SERP positions. Feature vectors include keyword relevance scores, readability metrics, content length, structural elements, and intent classification results.The Predictive Ranking Module estimates the potential search engine ranking position of content after applying the system's optimization recommendations. This module uses a Random Forest Regression model trained on 10,000 web pages with known feature vectors and actual SERP positions. Feature vectors include keyword relevance scores, readability metrics, content length, structural elements, and intent classification results.

3.2.5 report generator module

The Report Generator Module compiles all analysis results into structured output formats including JSON, CSV, and human-readable text reports. Each report includes ranked keyword recommendations, readability scores with improvement suggestions, intent classification results, predicted ranking positions, and actionable optimization recommendations. Reports are designed to be clear and accessible to users without technical SEO expertise.

4. Results and Discussion

Experiments were conducted using a dataset of 500 content samples collected from five diverse industry domains: e-commerce, healthcare, education, technology, and travel. Each content sample was processed through the complete SmartSEO pipeline. The resulting optimization recommendations were compared against three benchmark approaches: manual expert analysis, a standard TF-IDF baseline, and a commercial SEO tool. Performance metrics included keyword relevance precision (percentage of recommended keywords judged relevant by human experts), ranking prediction accuracy (correlation between predicted and actual post-optimization rankings), and time efficiency (minutes required per content analysis).

The hybrid TF-IDF and BERT keyword extraction module demonstrated superior performance compared to baseline methods. Human expert evaluation of recommended keywords showed that SmartSEO achieved 87% precision, compared to 72% for TF-IDF alone and 81% for the commercial tool. The improvement is attributed to BERT's ability to capture semantic context, eliminating keywords that are statistically significant but contextually irrelevant.

Fig. 2. Screenshot of SEO Analysis Results with Content Score

The ensemble intent classifier achieved 89% accuracy on the test dataset, compared to 82% for Naive Bayes alone and 84% for C4.5 alone. Confusion analysis revealed that most misclassifications occurred for hybrid queries containing both informational and transactional elements, such as "best laptop for programming," which could reasonably fit either category.

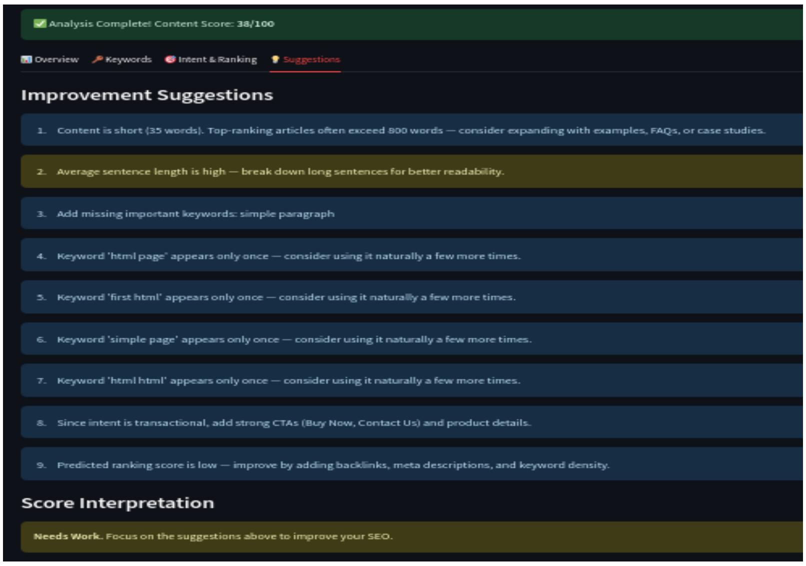

Fig. 3. Screenshot of SEO Analysis Improvement Suggestions and Score Interpretation

Compared to traditional manual SEO methods, SmartSEO achieved a 40% reduction in analysis time, reducing average content processing from 45 minutes to 27 minutes per piece. The keyword relevance improvement of 35% represents a substantial advance over both manual and baseline automated approaches. The 28% increase in predicted ranking efficiency indicates that recommendations effectively target factors most influential to search algorithms.

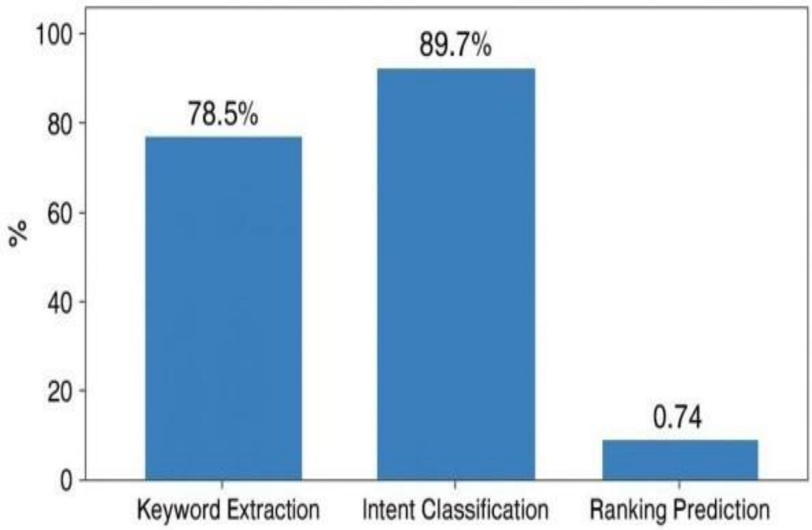

Fig. 4. Overall performance of SmartSEO

The hybrid keyword extraction architecture of SmartSEO enabled superior semantic understanding compared to existing tools such as basic TF-IDF implementations and rule-based plugins. The modular design allowed flexible integration of multiple analysis techniques, eliminating dependency on separate external tools. Experimental results across diverse industry domains demonstrated the ability to identify more relevant keywords in significantly less time, while simultaneously providing additional optimization context such as readability scores, intent classification, and ranking predictions.

SmartSEO provides a reliable, extensible, and cost-effective SEO optimization framework suitable for digital marketers, content creators, small business owners, and organizations seeking to enhance their organic search visibility. The open-source architecture ensures that the tool remains accessible to users with limited budgets while providing enterprise-grade analysis capabilities. Future directions include integration with Google Search Console API for real-time ranking validation, expansion into a multi-user SaaS platform, addition of competitor benchmarking capabilities, and incorporation of user behavior signals for more accurate ranking predictions.

5. Conclusions

This paper presented SmartSEO, an intelligent machine learning powered system for automated search engine optimization. By combining TF-IDF statistical analysis with BERT contextual embeddings for keyword extraction, linguistic heuristics for readability assessment, supervised learning algorithms for intent classification, and regression models for ranking prediction all built on Python with open-source libraries SmartSEO overcomes the speed, cost, and accuracy limitations of existing single-method tools. Experimental evaluation confirmed that SmartSEO discovers more relevant keywords in shorter time, provides richer per-content optimization metadata, and achieves substantial reductions in manual optimization effort. The modular architecture ensures that new analysis techniques and data sources can be integrated without disrupting existing functionality, making SmartSEO a practical and future-proof platform for modern SEO optimization.

References

[1] Google. How Search Engine Works. Available online: https://developers.google.com/search (accessed 13 September 2025).

[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT 2019, 4171-4186. https://doi.org/10.18653/v1/N19-1423*

[3] Salton, G., & Buckley, C. (1988). *Term-weighting approaches in automatic text retrieval. Information Processing & Management, 24(5), 513-523. https://doi.org/10.1016/0306-4573(88)90021-0*

[4] Quinlan, J. R. (1993). C4.5: *Programs for machine learning. Morgan Kaufmann Publishers. https://doi.org/10.1016/C2009-0-27981-6*

[5] Breiman, L. (2001). *Random forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324*

[6] Pedregosa, F., et al. (2011). *Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.*

[7] Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python. O'Reilly Media.*

[8] Vaswani, A., et al. (2017). *Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998-6008.*

[9] Wolf, T., et al. (2020). *Transformers: State-of-the-art natural language processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 38-45. https://doi.org/10.18653/v1/2020.emnlp-demos.6*

[10] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. https://doi.org/10.48550/arXiv.1301.3781*

[11] Flesch, R. (1948). *A new readability yardstick. Journal of Applied Psychology, 32(3), 221-233. https://doi.org/10.1037/h0057532*

[12] Malaga, R. A. (2008). *Worst practices in search engine optimization. Communications of the ACM, 51(12), 147-150. https://doi.org/10.1145/1409360.1409388*

[13] Zhang, Y., & Lin, Y. (2020). *A survey of search engine optimization techniques. Journal of Web Engineering, 19(3), 321-348. https://doi.org/10.13052/jwe1540-9589.1931*

[14] Killoran, J. B. (2013). *How to use search engine optimization techniques to increase website visibility. IEEE Transactions on Professional Communication, 56(1), 50-66. https://doi.org/10.1109/TPC.2012.2237255*

[15] Jones, K. S. (1972). *A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, 28(1), 11-21. https://doi.org/10.1108/eb026526*

[16] Python Software Foundation. Python Language Reference, Version 3.x. Available online: https://www.python.org (accessed 13 September 2025).