Anusandhanvallari
Vol 2024, No.1
December 2024
ISSN 2229-3388

Information Extraction with Semantic Term Relation Learning combined with Document Classification using Ranking and Trust Assessment

Rathan Kumar Chenoori[1], Sunil Kumar Thota[2], Pillareddy Vamsheedhar Reddy[3], Padma BalaKrishna[4]

1,2 Assistant Professor, Department of CSE, Keshav Memorial Institute of Technology, Hyderabad, Telangana, India

3 Assistant Professor, Department of CSE-AIML, Keshav Memorial Engineering College, Hyderabad, Telangana, India

4 Assistant Professor, Department of CSE, Keshav Memorial Engineering College, Hyderabad, Telangana, India

Abstract: In the era of abundant information, getting efficient and correct information from unstructured documents is of highest importance. This paper proposes a detailed method for information extraction, which includes learning of relationships between terms using semantics and classifying documents with ranks and scores of trust. The method uses semantic analysis to understand the relationships between terms in the documents and using natural language processing methods, the model identifies and extracts relationships between terms. The relationship between the terms act as indications for understanding the context and retrieving the information from the documents. The agent uses neuro-fuzzy rules in order to deal with time-based data; a deep neural network that is based on semantic similarity then sorts this data by semantic relationships. The system looks at document features and the semantic connections between terms for classification. A ranking system scores each document by relevance and trustworthiness to increase usability. The ranking is based on things like the source's reliability, past accuracy, and user input. This makes sure that users see the most relevant and reliable documents, which improves their experience and trust in the system. This method helps increase the correctness and reliability of information extraction systems, allowing users to access useful and trustworthy information from many unstructured documents.

Keywords: *Semantic Similarity, LSTM Networks, Convolution Neural Network, Deep Learning, Neuro-Fuzzy*

1. INTRODUCTION

This study looks at ways to find information in online content, which has greatly increased. It concentrates on getting useful information from large web datasets using text mining. The main goal is to create text classification systems that can accurately label documents. These systems should improve how digital information from different sources is organized and found. The research assumes that standard data processing can't handle the amount and complexity of current web information. Modern applications, like financial forecasting and sentiment analysis, need strong classification systems to turn unstructured text into analyzable formats for processing and decision-making.

The paper focuses on semantic analysis to understand the meaning of words in online and social media discussions. Semantic links are measured using similarity scores based on how close words are to each other. Semantic distance measures the difference between ideas, giving insight into how terms relate. The study uses WordNet to improve semantic understanding and extracts semantic patterns from large web datasets to better interpret content and enhance information retrieval.

The research divides feature selection methods into three types: filter-based, wrapper-based, and embedded. This study enriches the process of text classification through the incorporation of temporal element extraction, which accounts for the shifts in trends and sentiments as they develop. This method of classification is very helpful

because it gets the subtle changes in both language and setting. By mixing features that are both local and global, we see an improved capacity to notice how text changes with time.

Our research takes advantage of text summarization through assessments of word and phrase frequency. The system uses methods of sentence choice and summary creation. For classification, deep learning models such as LSTM networks have been adopted. A convolutional neural classifier is a main part of the work that pushes forward the ability to extract useful info from big sets of data. The data shows that the system has improved accuracy when measured against current methods across different platforms and web content.

2. RELATED WORK

Work in semantic text analysis has advanced in feature selection, document summarization, and classification. In [1] document categorization using a pattern mining algorithm via closed frequent itemsets. Their method improved classification accuracy and query response time. Veloso et al. [2] created a ranking system for web documents using rule mining from training data to find similarity scores based on word associations.

Babashzadeh et al. [3] introduced concept association rule mining (ARM), treating user queries as conceptual entities and using ARM to raise classification accuracy and query relevance. Brown et al. [5] put forward a feature selection plan based on information theory that did better than older ways.A supervised and semi supervised selection of features model that kept inter-class correlations while selecting attributes based on query relevance was proposed by Wang et al.[5].Braytee et al.[6] built upon this work by using multilabel-based nonnegative matrix factorization, improving document relevance across online sources.

Zhang et al. [7] combined topic relevance using inter-term correlation values to find contextually relevant content from local repositories, addressing large-scale data management while keeping retrieval accuracy. Rao et al. [8] used a two layered neural network where the first layer assessed sentence-level semantic relevance with vector representations, and the second encoded inter-sentence relationships for document representation. Their system included preprocessing such as input cleaning and emotional polarity filtering, and outperformed baseline models across three review datasets.

Altinel and Ganiz[9] compared semantic and traditional text classification, organizing methods into five groups namely domain knowledge driven, corpus based, deep-learning, sequence-models, linguistic methods. They found that semantic methods offer better classification. Sinoara et al. [10] developed a semantic representation framework for web documents with embedded term and term-sense vectors with word sense disambiguation to create rich semantic features and low-dimensional representations.

Srivastava et al. [11] designed a machine learning framework using two medical tweet datasets and the WebKB4 dataset. Their model used random forest, SVM with radial-basis functions, and a multilayer perceptron, reaching 97% accuracy and showing gains through incremental feature selection. A sentiment analysis classifier based on semantic similarity scores from embedding techniques, improving upon benchmarks was introduced byGoularte et al. [12] proposed a text assessment model that used fuzzy logic to assess extracted features and make abstractive summaries. Their system found key content and showed correlation-based performance when compared with expert summaries, reducing semantic summarization tasks.

He et al.[13] introduced a convolutional neural network with multipooling operations to classify medical records with multidirectional features. Zhang et al. [14] developed a triple way enhanced CNN ensemble design that improved classification accuracy more than standard CNN approaches. A Hebb rule-oriented feature selection method that treated terms and classes as neurons, identifying words that excited related classes, and showed superior performance was created by Heyong and Ming [15]. Perumal et al. [16] developed a fuzzy-rule based electronic learning recommendation system for interests of users that gave content adapted to user needs. Camastra

et al. [17] created graphical document presentations with semantically meaningful information and visual formats, using semantic maps for better accessibility.

Wu et al. [18] introduced sentiment-based summarization with emotion modeling, showing efficiency in text summarization with emotion. Yang et al.[19] presented adversarial networks to generate informative summaries with grammar using networks trained by adversarial learning, giving enhanced generative models through multitask learning.

Ma et al. [20] advanced deep feature learning with methods that stressed feature analysis, improving detection accuracy for semantic data requiring feature understanding.He et al. [21] used a CNN to classify medical records. Despite gains, many models struggle to meet requirements for accuracy, mainly with social media data. Limitations show the need for systems with better predictive and classification performance, and greater contextual relevance in semantic analysis and text classification.

3. SYSTEM FRAMEWORK

The proposed model operates as per the structure presented in Fig. 1. The architectural design encompasses several pivotal elements, including the user query request module, preprocessing module, Semantic Learning and Summarization Module, the classification module, Ranking and Trust Calculation Module components.

User Query Request and Preprocessing Module:

This module manages initial user interactions by capturing query inputs and executing structured preprocessing steps. User queries are tokenized to extract keywords, followed by refinement via removal dictionaries that eliminate generic or non-informative terms. These processed keywords serve as input for web content extraction routines, enabling the retrieval of relevant documents and establishing the foundation for subsequent semantic and classification procedures.

Semantic Learning and Summarization:

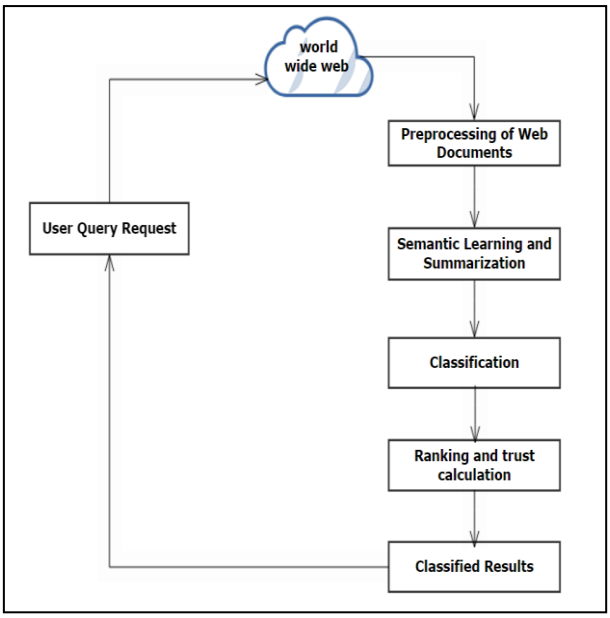
After data is retrieved, key terms are selected and an algorithm based on clustering groups documents by semantic-similarity and then condenses the content into summaries that are coherent and relevant which keeps essential info and reduces data size for processing.

Classification:

The summarized content is classified with a combination of deep neural-networks and analysis of semantics. For accurate categorization of the documents a neuro-fuzzy logic refined progressively by temporal feature is used. For improvement of classification in complex data sets in semantic and temporal relationships a glance of multiple layers are made.

Ranking and Trust:

Ranking and Trust computation in decision making and information retrieval is used to find reasonable content while eliminating extraneous or untrustworthy sources. Ranking organizes items by specific standards, typically from most to least important. Trust calculation is a method of judging how reliable a source, person, or piece of info is. which takes into account things like reputation, past actions, and confirmation to decide how much faith to put in that source or info.



4. PROPOSED WORK

The paper presents a comprehensive information retrieval system which combines several methods involving choosing characteristics based on similarity of semantics, data summarization, classifying of information using a deep neural-network combined with text-based relational analysis, and adding progressive time-related features using neuro-fuzzy rule-based agents. These parts work together in one algorithm which improves the information retrieval process.

4.1 Feature Selection based on Semantic Analysis

The feature selection mechanism carries out on semantic similarity scores evaluated from web-extracted terms. The approach treats words as comprising meaning and structure during embedding and classifying them as trained terms that remain unbiased incorporating additional data sources.

The text segments are defined mathematically as:

$$W^{i}=\{w^{i}_{1,}w^{i}_{2,}w^{i}_{3,}w^{i}_{4,}w^{i}_{5,}\ldots.,w^{i}_{j,},\ldots..w^{i}_{J}\} \quad (1)$$

Here, 'j' indicates text portions made up of input tokens, and these portions can be anything from single sentences to entire online documents.

The meaningful terms are grouped by the system into lexicons derived from terminology related to sentiment vocabulary, where each term carries a specific polarity value. The target sentiment terms are defined through:

$$W^{st}=\{w^{st}_{1,}w^{st}_{2,}w^{st}_{3,}w^{st}_{4,}w^{st}_{5,}\ldots.,w^{st}_{j,},\ldots..w^{st}_{LEN}\} \quad (2)$$

Similarity score calculation makes features for each word by figuring out how alike input terms and sentiment terms are.

$$S_{i,j}=simi\_score\,(w_j^{(i)},w_j^{s}) \quad (3)$$

Where $S_{i,j}\in[0,1]$, with values of $w_j^{(i)}$ and w_j^{s} indicating similarity levels. The pooling function operates column-wise on the similarity score matrix:

$$P_j = maxsim(w_k^{(i)}, w_j^{(s)}) \quad \text{for k } \in \{1, 2,..,I\} \qquad (4)$$

Figure 1 System Architecture

The SIMON algorithm uses WordNet semantic similarity and embedding-oriented term similarity scores, combining them through dot product operations.

$$\text{sim}(w_i^{(i)}, w_j^{(s)}) = E_{w_i^{(i)}}^T E_{w_j^{(s)}} \qquad (5)$$

Algorithm: Semantic-Similarity Score and SIMON value-based Feature Extraction
Input: Terms extracted from Web
Output: Features combined with weights in vector v

1. **Define** W_i as input terms set: $W^i = \{w_{1,}^i w_{2,}^i w_{3,}^i w_{4,}^i w_{5,}^i \ldots., w_{j,}^i, \ldots.. w_J^i\}$
2. **Define** SL as sentiment lexicon
3. **Word Selection** from SL: $W^{st} = \{w_{1,}^{st} w_{2,}^{st} w_{3,}^{st} w_{4,}^{st} w_{5,}^{st} \ldots., w_{j,}^{st}, \ldots.. w_{LEN}^{st}\}$
4. Calculate sentiment scores: $l = [l_1, l_2, ..., l_{len}]$
5. **For each** term w_i^(i) in W(i):
6. **For each** term w_i^(s) in W(s):
7. **Calculatesimilarity score:** $S_{i,j} = \text{simi\_score}(w_i^{(i)}, w_j^{(s)})$
8. **For k in range 1 to I:**
9. **Compute** feature vector: $Pj = \max(S_{:j}) = \max(\text{sim}(w_k^{(i)}, w_j^{(s)}))$
10. **Calculate** sentiment weighting: $v = l \times p$

Feature extraction is conducted through a two-stage process. First, terms are filtered according to their frequency in the input data using predefined thresholds. To evaluate the relationship between the selected features and target labels ANOVA tests are applied. For every feature in the dataset the F-measure scores are computed, where the degree of feature informativeness determine the final selection.

4.2 The Embedding Procedure to Represent Text

The Lexicon Method-Based Selection embedding process changes input data into particular feature lengths using text-based embedding representation.:

1. W(st), l = select(SL, dataset)
2. tmp = Freq_Filter(SL, dataset)
3. W(s), l = anova_filter(tmp, dataset)

The vector values are taken from each piece of data, and then average pooling is done on those values to streamline the terms and data. Refinement of sentiment analysis models in information retrieval is done by applying paragraph-based vectorization to lengthy texts which distinguishes between texts of different lengths.

4.3 Improved Incremental Feature Expansion

To observe the impact on data retrieval an approach where the features are added incrementally is applied. Two sets of data are employed train classifiers like EM-SVM, Naïve-Bayes, and Decision- Tree Induction. Several relevant features are identified with their base labels and relevant feature sets from vast number of web e-documents as input. The input data coefficient values computed by the methods transform testing datasets into online predictors of classes. This ensures attributes are carefully chosen from the above specified classifiers. The algorithms also increase the segregation index value based on empirical data while handling factors like filtering, variance and complexity. The Bag of Words (BOW) model serves as a straightforward way to represent documents which treats a text or web document as just a group of words. It classifies input text, using frequency of occurrence of certain terms as important features during training phase. It is derived from T_Freq-ID_Freq, which gives a word high weight. To compute the weight T_Fre-ID_Fre we need frequency in the corpus referred ast_f. The method to calculate T_Freq is in Equation (6).

$$T\_Freq(\mathrm{t,d}) = 1 + log(ft, d) \qquad (6)$$

whereft,d represents the term ‘t’ in the record ‘d’ .

The inverse document replication process has been determined using Eq. (7).

$$ID\_Freq(\mathrm{t,d}) = \log(1 + N/nt) \qquad (7)$$

Where nt represents the quantity of records contains the term ‘t’. T_Fre-ID_Fre is acquired by using Eq. (8).

$$T\_Freq - ID\_Freq(t, d, D) = T\_Freq(\mathrm{t,d}) \,.\, ID\_Freq(\mathrm{t,d}) \qquad (8)$$

This methodology identifies and eliminates unimportant words in large datasets, improving classification results by focusing on terms providing discriminative power.

Latent Semantic Indexing (LSI) links words to content concepts. It explores document-word relationships, finding concepts in online documents. LSI uses cosine similarity on normalized vectors within the documents. Similarity ranges from 1 (identical) to 0 (dissimilar). Query vector comparison uses LSI representation.

$$\mathrm{qvk} = \sum_{k}^{-1} U_k^T \,.\, \mathrm{qv} \qquad (9)$$

where "k" represents dimensions used to model standard and original texts, "k" refers to diagonal matrix inverse, and U_k^T represents matrix transpose multiplied by "k" to obtain singular vector U_k.

4.4 Summarizing Text with Semantics

This text summarization method uses fuzzy logic to produce useful document abstracts. It focuses on understanding meaning instead of just using statistics. The process has five steps that turn raw text into summaries by finding important features and judging how relevant they are.

4.4.1 Preprocessing of Text

Preprocessing is key to good summarization, starting with dividing text into sentences and then into tokens. Next, stemming reduces words to their root form, like changing lying to lie. This cuts down on data but keeps the meaning clear, helping with later analysis.

4.4.2 Evaluating words and sentences using semantic-based methods.

The scoring method looks at if words and full sentences make sense in relation to the topic. A word (T) includes some sentences: T = {S_1, S_2, S_3,...S_j}. J is how many sentences there are. Each sentence (S_k) contains a number of words (I): S_k = {t_1, t_2,...t_i}. The system uses different math formulas to score things.

Term Frequency Analysis:

$$TF(t,d) = \frac{f(t,d)}{\Sigma f(w,d)} \qquad (10)$$

Semantic Similarity Score:

$$SimScore(S_1,S_2) = \frac{\Sigma(w_{1i} \times w_{2i})}{\sqrt{(\Sigma {w_{1i}}^2)} \times \sqrt{(\Sigma {w_{2i}}^2)}} \qquad (11)$$

Position Weight Factor:

$$PosWeight(t) = log\left(\frac{TotalTerms}{Position(t)}\right) \qquad (12)$$

Length Relevance Measure:

$$LengthScore(S) = \frac{\text{OptimalLength}}{|\text{ActualLength} - \text{OptimalLength}|} \qquad (13)$$

(13)

Fuzzy analysis divides informativeness into four groups: low, medium, high, and very high. This gives a more detailed analysis than older three-level methods. Fuzzy functions check terms, word use, semantic scores, and total informativeness to keep data safe, which can be lost in simple yes/no setups.

4.5 Deep Learning and Neuro-Fuzzy Classification with Temporal Features

The classification system integrates deep neural networks with LSTM and GRU layers to process temporal dependencies and complex patterns. The function hidden-state is expressed as:

$$h_{st} = f(W.h_{st-1} - 1 + U.x_{st} + b) \qquad (14)$$

where,

- h_{st} represents the hidden-state at time step t
- x_{st} denotes the input at a time-step t
- W and U represent the weight matrices
- b represents the bias-vector

Advanced semantic understanding has been achieved through BERT models and topic-modeling techniques.

$$h_{st} = f(W.h_{st-1} - 1 + U.x_{st} + b) \quad (15)$$

- Word w's probability in document d is represented by P(w/d)
- Word w's probability for a given topic t is indicated by P(w/t)
- Topic t's probability given document d is shown by P(w/t)

The intelligent agent-based neuro-fuzzy system combines neural network outputs with fuzzy rule generation. Triangular membership functions define fuzzy set boundaries:

Where parameters m, n, and p define the triangular function shape, with membership values ranging from 0 to 1

$$\mu(x) = \max\left(0, \min\left(\frac{x-m}{n-m}, \frac{p-x}{p-n}\right)\right) \quad (16)$$

Temporal pattern recognition utilizes Recurrent Neural Networks and Temporal Convolutional Networks to identify time-based trends. The RNN temporal processing function:

$$h_t = \tanh(W_h * h_{t-1} + W_x * X_t) \quad (17)$$

Where W_h and W_x represent weight matrices for hidden states and inputs respectively.

4.6 Ranking and Trustworthiness Calculation

The Trust-based Rank algorithm evaluates web page credibility through iterative trust propagation from manually selected seed pages. The algorithm incorporates a decay factor α to control trust diminishment rates during propagation, with higher α values indicating faster trust decay. The complete algorithm processes as follows:
Input Parameters: TM: transition matrix, IM: inverse transition matrix, N: total pages, L_i: invocation limit, α: decay factor, M_i: iteration count, h_{is}: inverse HostRank scores

Step 1: Initialize Desire Seeds, $h_{is} = 1/N$
Step 2: Iterative Score Update,
for i = 1 to M_i:
$h_{is} = \alpha \times IM \times his + (1-\alpha) \times (1/N) \times 1_n$
Step 3: Ranking Calculation, $\sigma = Rank(\{1,...,N\}, h_{is})$
Step 4: Distribution Score Application,
$ds = 0_n$
for i = 1 to L_i:
if $O(\sigma(i)) == 1$:
$ds(\sigma(i)) = 1$
Step 5: Normalization, $ds = ds / |ds|$
Step 6: Final Trust Score Computation,

ts* = ds
for i = 1 to M_i:
ts* = α × TM × ts* + (1-α) × ds

5. RESULTS AND DISCUSSION

A Java-based document retrieval system using semantic analysis was built with the J2EE framework. The system processes web content via preprocessing, summarization, and text classification to match user queries with pertinent info. Its performance was tested in web-based and local environments using info retrieval metrics. Three datasets—Amazon product reviews, Multidomain Sentiment (historical Amazon reviews from Johns Hopkins), and Tweets Debate—validated the sentiment analysis model across text types.

5.1 Evaluation Metrics

The system's effectiveness was quantified through three fundamental measurements. Precision assessment determines the proportion of relevant content within retrieved results as, Precision, Recall, F-Measure. Along with these used Cohen's Kappa (CK) statistic enhances model accuracy by grouping terms with comparable semantic weights.

Number of Web Documents	Without Semantic Similarity	With Semantic Similarity
E1	98.55	98.6
E2	98.6	98.7
E3	98.7	98.9
E4	98.75	99.2
E5	99.1	99.4

Table-1: Performance Analysis

This reliability measure operates within the 0-1 range, where 0 indicates random similarity and 1 represents perfect correspondence:

$$CK = 1 - \frac{1 - po}{1 - pe} \quad (18)$$

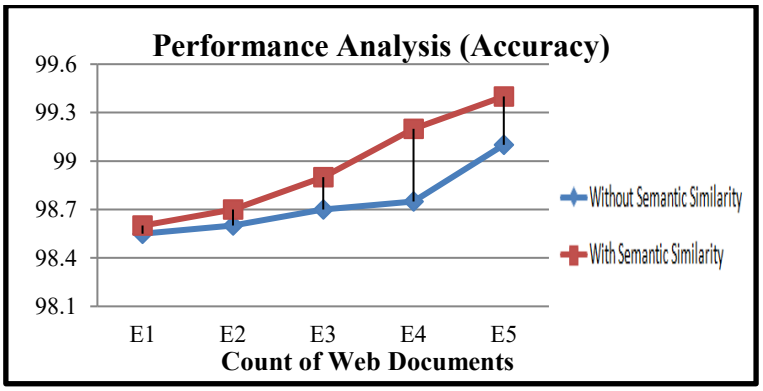


Figure 2 Performance Analysis (Accuracy) based on number of documents with and without semantic similarity

Where po represents semantically relevant and meaningful terms, while pe denotes the probability of semantic relevance. The CK interpretation follows defined ranges: (0-0.3), (0.3-0.6) low-medium, (0.6-0.8) (medium), and (0.8-1.0) high semantic similarity levels.

5.2 Experimental Results

Five experiments (E1-E5) checked how well preprocessing worked, how good the semantic similarity summarization was, and how accurate the classification predictions were.

Table-2: Accuracy Analysis of Relevancy

Percentage of Documents	Proposed Model + Semantic Similarity without Text Summarization	Proposed Model without Semantic Similarity Score	Proposed Model with Semantic Similarity Score	Proposed Model + Semantic Similarity + Text Summarization
20	99.3	98.5	98.7	99.4
40	99.35	98.6	98.8	99.5
60	99.4	98.65	98.9	99.6
80	99.45	98.7	99.05	99.65
100	99.46	98.8	99.1	99.8

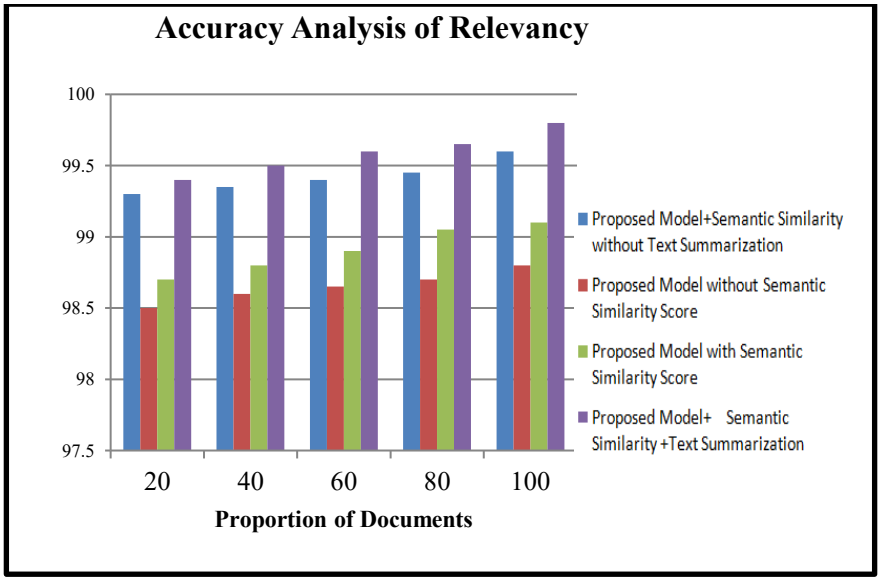


Figure 3 Relevancy Score Analysis with and without consideration of text summarization

Web content datasets had 200 to 1000 documents, with text inputs from 20% to 100%. Using semantic similarity made relevancy accuracy better. Figure 2 showed that adding semantic similarity values improved the model for all document sizes.

Table-3: Analysis of Relevancy Score

Experiments	SVM + Semantic Similarity + Text Summarization	FTCM + Semantic Similarity + Text Summarization	CNN + Semantic Similarity + Text Summarization	Deep Learning + Neuro Fuzzy Classification + Semantic Similarity + Text Summarization
E1	98.5	99.3	99.4	99.5
E2	98.6	99.4	99.4	99.6
E3	98.7	99.45	99.6	99.6
E4	98.8	99.5	99.65	99.7
E5	98.8	99.6	99.7	99.8

Semantic analysis-based text summarization was better than older methods because of smart semantic scoring in decisions. Figure 3 and Table 2 showed the semantic analysis model worked better than methods without similarity or summarization. Semantic analysis helped find relevant documents better in all experiments, improving on baseline methods.

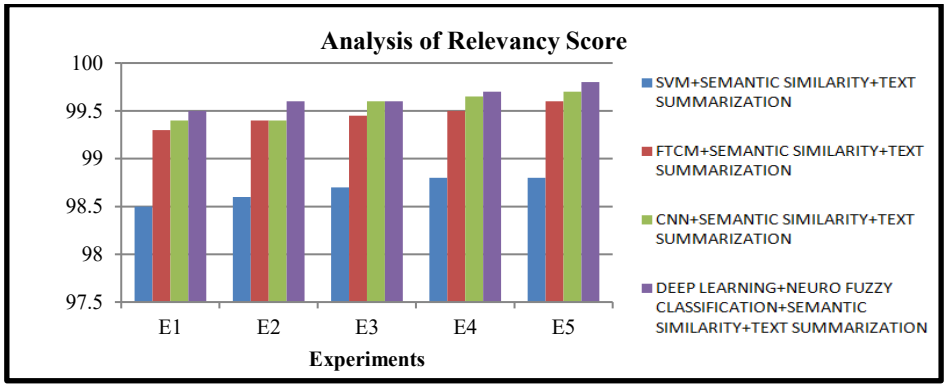


Figure 4 Relevancy score analysis between the proposed and existing systems

Figure 4 and Table 3 showed models using both semantic similarity and summarization did better than using each alone. The five experiments confirmed strong performance in different situations.

Table-4: Accuracy Analysis of Prediction

Experiments	Deep Learning + Neuro Fuzzy Classification + Semantic Similarity	Deep Learning + Neuro Fuzzy Classification + Semantic Similarity with

	Without Text Summarization	Text Summarization
E1	99.20	99.40
E2	99.30	99.45
E3	99.40	99.65
E4	99.45	99.70
E5	99.60	99.85

Figure 5 and Table 4showed the new information retrieval model worked better than old systems without semantic analysis or summarization. Different relevance levels showed how much relevant information each document had, with semantic analysis summarization giving steady benefits.

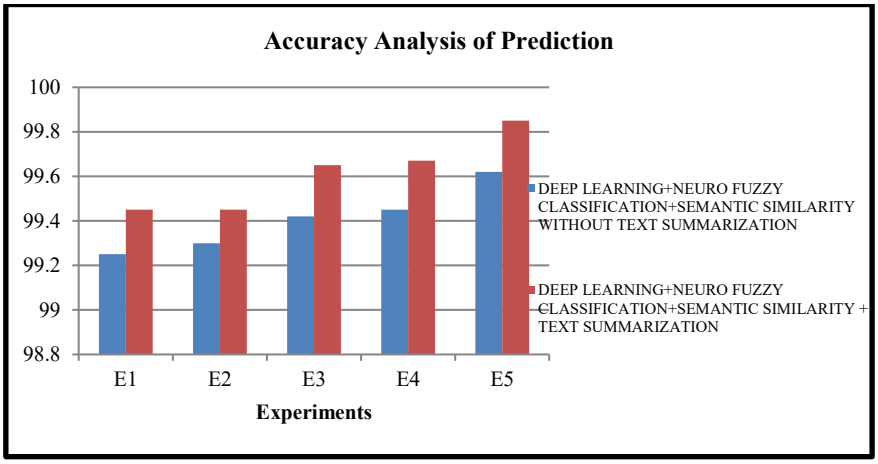


Figure 5 Prediction accuracy analysis between the proposed models with various sizes of documents

6. CONCLUSION

This paper introduces a new model for finding relevant data. The model uses semantic analysis to get important info from large databases or online applications. It has parts like a feature selection process using semantic similarity, a data summarization method, and a classifier using deep learning and neuro-fuzzy methods that considers text relationships. The new algorithm for feature selection helps find and take out similar data from local and online sources. The semantic summarization technique shortens web info. A deep learning and neuro-fuzzy classifier sorts data based on semantic relationships and also calculates trust and ranks documents. Tests show the model works well for data retrieval, pulling relevant data from online sources and datasets. Future work could include integrating Temporal Convolutional Networks with neuro-fuzzy systems, combining Recurrent Neural Networks with fuzzy logic, or making hybrid models that mix statistics with deep learning and fuzzy logic to improve classification.

REFERENCES

[1] Yu H, Searsmith D, Li X, Han J (2004) Scalable construction of topic directory with nonparametric closed termset mining. In: Proceedings of the fourth IEEE international conference on data mining. IEEE, pp 563–566

[2] A. A. Veloso, H. M. Almeida, M. A. Gonzalves, and W. MeiraJr, "Learning to rank at query-time using association rules," in Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr., ACM, 2008, pp. 267–274.
[3] A. Babashzadeh, M. Daoud, and J. Huang, "Using semantic-based association rule mining for improving clinical text retrieval," in Health Information Science, G. Huang, X. Liu, J. He, F. Klawonn, and G. Yao, Eds. Berlin: Springer, 2013, pp. 186–197.
[4] G. Brown, A. C. Pocock, M. Zhao, and M. Luján, "Conditional likelihood maximization: a unifying framework for information theoretic feature selection," J. Mach. Learn. Res., vol. 13, pp. 27–66, 2012.
[5] J. Wang, J. Wei, and Z. Yang, "Supervised feature selection by preserving class correlation," in Proc. 25th ACM Int. Conf. Inf. Knowl.Manag., Indianapolis, USA, Oct. 2016, pp. 1613–1622.
[6] A. Braytee, W. Liu, D. R. Catchpoole, and P. J. Kennedy, "Multi-label feature selection using correlation information," in Proc. ACM Conf. Inf. Knowl. Manag., Singapore, Nov. 2017, pp. 1649–1656.
[7] Zhang J, Li C, Cao D, Lin Y, Su S, Dai L, Li S (2018) Multi-label learning with label-specific features by resolving label correlations. Knowl Based Syst 159:148–157
[8] G. Rao, W. Huang, Z. Feng, and Q. Cong, "LSTM with sentence representations for document-level sentiment classification," Neurocomputing, vol. 308, pp. 49–57, 2018.
[9] B. Altınel and M. C. Ganiz, "Semantic text classification: a survey of past and recent advances," Inf. Process. Manag., vol. 54, pp. 1129–1153, 2018.
[10] Sinoara RA, Camacho-Collados J, Rossi RG, Navigli R, Rezende SO (2019) Knowledge-enhanced document embeddings for text classification. Knowl-Based Syst 163:955–971
[11] Srivastava SK, Singh SK, Suri JS (2019) Effect of incremental feature enrichment on healthcare text classificationsystem: a machine learning paradigm. Comput Methods Programs Biomed 172:35–51
[12] F. B. Goularte, S. M. Nassar, R. Fileto, and H. Saggion, "A text summarization method based on fuzzy rules and applicable to automated assessment," Expert Syst. Appl., vol. 115, pp. 264–275, 2019.
[13] B.He, Y. Guan, and R. Dai, "Classifying medical relations in clinical text via convolutional neural networks," Artif.Intell.Med., vol. 93, pp. 43–49, 2019.
[14] Y. Zhang, Z. Zhang, D. Miao, and J. Wang, "Three-way enhanced convolutional neural networks for sentence-level sentiment classification," Inf. Sci., vol. 477, pp. 55–64, 2019.
[15] W. Heyong and H. Ming, "Supervised Hebb rule-based feature selection for text classification," Inf. Process.Manag., vol. 56, pp. 167–191, 2019.
[16] S. P. Perumal, G. Sannasi, and K. Arputharaj, "An intelligent fuzzy rule-based e-learning recommendation system for dynamic user interests," J. Supercomput., vol. 75, no. 8, pp. 5145–5160, 2019.
[17] Camastra F, Ciaramella A, Maratea A, Son LH, Staiano A (2020) Semantic maps for knowledge management of web and social information. In: Acampora G, Pedrycz W, Vasilakos A, Vitiello A (eds) Computational intelligence for semantic knowledge management. Studies in computational intelligence, vol 837. Springer, Berlin
[18] Wu P, Li X, Shen S, He D (2020) Social media opinion summarization using emotion cognition and convolutional neural networks. Int J Inf Manag 51:101978
[19] Yang M, Wang X, Lu Y, Lv J, Shen Y, Li C (2020) Plausibility promoting generative adversarial network for abstractive text summarization with multi-task constraint. Inf Sci 521:46–61
[20] W. Ma, Y. Wu, F. Cen, and G. Wang, "MDFN: multi-scale deep feature learning network for object detection," Pattern Recogn., vol. 100, p. 107149, 2020.
[21] He B, Guan Y, Dai R (2019) Classifying medical relations in clinical text via convolutional neural networks. ArtifIntell Med 93:43–49