ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 1 - 2026

Deep integration of artificial intelligence and big data technologies in the news production process

Ning Shuo[1,2,*]

[1] Television School, Communication University of China, Beijing, 010024, China
[2] College of Liberal Arts & Journalism, Inner Mongolia University, Hohhot, Inner Mongolia, 010021, China

SUMMARY: *News text summary generation is an important part of the news production process, which is based on the news text keywords to highly condense the text content and help the news information to be quickly understood and smoothly digested. This paper combines the Htmlparser tool with an improved template-based SST tree method to obtain key news information text in web pages. Aiming at the characteristics of large data volume and more isolated points of news event information, the data metaclustering algorithm under the idea of density clustering is proposed for feature mining of news information text. On this basis, the LDA topic model is used to obtain the topic difference influence degree of information text keywords, the fusion of LSTM model and word2ec model is used to calculate the semantic relevance influence degree, and the two influence degree features are put into the TextRank algorithm to calculate the keyword importance. A generative code summarization model structured by multi-component encoder, encoder and keyword guidance is formed, and the keyword information is used to guide the encoder to generate summaries corresponding to the topics of the original text, so as to transform structured text data into short code summaries. The keyword-guided news generative summarization model generates new word rates between 18.97% and 91.01% in different structural tasks, and the three evaluation indexes of ROUGE-1, ROUGE-2, and ROUGE-L are as high as 46.36%, 29.78%, and 41.49%, respectively. The automatic generation of abstracts with artificial intelligence supported by big data technology realizes the information summarization and framework refinement of news text, and promotes the double improvement of news production quality and efficiency.*

KEYWORDS: *generative summary model; data metaclustering; news production; artificial intelligence; keyword guidance*

1 Introduction

In the context of the era of informationization and intelligence, artificial intelligence (AI) and big data technology have changed the original news media pattern and ecology. The traditional news production model is facing a serious impact, and the three elements of news authenticity, timeliness, and accuracy are challenged under the rapid development of new media, and the demand for precision and other needs have become the key to news output at this stage, while AI and big data are precisely the new production model for news media to realize [1-4].

The large-scale popularization and application of AI and big data are reshaping the original news production mode step by step. Through intelligent algorithms, the news media can sift out

*lemonade1025@163.com
https://doi.org/10.65102/is2026249
Published: 30 April, 2026

useful materials from a large amount of information and automatically generate manuscripts, which dramatically improves work efficiency. Janáčková [5] used an AI-based text generator to write short radio readings for a radio program and to automate the generation of news texts by modifying and determining the final text based on the final text generation requirements. Tsourma et al [6] proposed an AI-based framework capable of retrieving a large number of Earth observation data requests for a specific event such as a disaster, and generating and modifying personalized news report content based on the user's style in real time. Pathak et al [7] designed a news summary generator based on natural language processing and machine learning to realize intelligent generation of news summaries with contextual meaning under the process of text preprocessing, keyword extraction and semantic understanding. Guo et al [8] investigated that the Internet and big data technologies have accounted for more than 43% of commercial news production by mining valuable information, shifting the main body of news production from a single subject to multiple subjects. He [9] emphasized that with the help of big data-based charts and graphs, by analyzing the news as a whole, it is possible to generate a variety of presentations of news information and enhance the readability of the news, as well as analyze the user needs and preferences and feedback information to make news adjustments.

Ai and Li [10] proposed a hybrid recommendation method based on deep learning and preference feature recommendation algorithms for big data for accurate personalization of news media content, which increased the click-through rate of news content by 20%. Altheneyan and Alhadlaq [11] proposed a stacked integration model and a stacked integration classification model based on big data machine learning techniques using distributed learning for fake news detection to avoid fake data in collecting news material. Compared with traditional manual editing, big data and AI can accurately capture audience interest points through the analysis of historical data and real-time hot trends, effectively promoting content creation toward automation and intelligence. In addition, the application of advanced technology has further accelerated media integration, and the simultaneous production of text, video and audio has gradually become normalized, effectively meeting the diversified needs of modern communication. O'Halloran et al [12] integrated multimodal frameworks, big data, AI tools, image/video processing, and textual metadata computational models into a multimodal analytics platform for multimodal news data analysis, which realizes diversified generation with the support of AI technology. However, the technology-driven change also puts higher requirements on the professional ability of related practitioners, and news media need to seek a balance between efficiency improvement and quality control.

Heim and Chan-Olmsted [13] found that consumers favor lower AI technology integration during AI-driven news production, but that consumer trust and intention to use increase in human-led AI-assisted news production models. However, Ojoajogwu et al [14] noted that the adoption of AI for fact-checking and validation tools for news content was high in the news generation process, but there were gaps in the application and training of employees on AI tools, which led to the ineffective use of AI. Pleios and Tastsoglou [15] summarized the key challenges of AI-generated content in news production, i.e., data incompleteness, lack of transparency of news sources, copyright and originality, and lack of critical approach. Therefore, how to find a position and achieve sustainable development in the midst of change is an urgent topic for news media based on AI and big data nowadays.

In this paper, Htmlparser tool is used to parse the information of news webpage, improve the body text extraction method based on template SST tree for information feature extraction, and then propose data metaclustering algorithm by combining with density clustering concept. After completing the extraction and mining of webpage information text, we synthesize TextRank algorithm, LAD topic model, LSTM model and Word2ec model, and establish the keyword importance calculation method containing word representation module, topic

differentiation module, semantic relevance calculation module and probability transfer matrix construction module. Under the task of structured text summary generation, the information encoding of context vectors, the input of decoder, and the keyword-guided summary generation are sorted out, and the news-generated summary model integrating structural and contextual information is constructed.

2 News information text extraction and clustering design

2.1 Extraction of news information text

2.1.1 Parsing of web pages

The parsing part of the web page is mainly done by Htmlparser open source toolkit.HtmlParser is a convenient and fast web page parsing tool which can easily extract the tags, tag attributes, attribute values and text values of the web page. And filtering the label information in the web page is one of the very important functions of HtmlParser. For the original web pages, due to the large number of web pages with different writing styles and the huge number of web pages. It is very troublesome to use regular expressions to match the content of the corresponding key fields. Therefore, the system utilizes the general interface provided by HtmlParser to extract some key contents of the web pages, such as web page titles, body titles, hyperlinks, anchor text and other information.

By establishing corresponding text filtering and link filtering conditions on the webpage, the corresponding link nodes and text nodes can be extracted from the original webpage. With the above method, the key information of the web page needed by the system can be obtained.

2.1.2 Extraction of web page body text

By parsing the web pages, it is possible to get the important elements of the web pages that are single and fixed. In order to effectively obtain the body of the web page, it is necessary to go through a more grammatical denoising algorithm in order to obtain a purer body of the web page. In this regard, this subsection uses an SST-based body text extraction algorithm and optimizes the algorithm for measuring the node weights in it. The basic assumption of this algorithm is that, in general, many web pages in the Web are very often similar in page structure, especially for pages in the same site. It is based on this assumption that the system is implemented to obtain a portion of web pages and analyze the commonalities and differences between them. Then the weights of each node in the DOM tree are calculated, and the weights are calculated using information entropy. This ensures that the greater the structural difference, the greater the entropy value of the information. Finally, the meaningless nodes in the web page are filtered by setting a threshold of node importance. The structure template for web page body extraction is obtained to extract the web page body.

The implementation details of the algorithm are as follows. Firstly, IHTMLDocument3 in the control mshtml is used to generate the DOM tree of each document. In general, 100 pages prepared in advance are utilized to build the corresponding Site Style Tree (SST).

The process of building the SST is a merging process of the DOM trees in the training samples. For all the children of a parent node in the DOM tree, if the corresponding child node exists in the corresponding parent node in the SST, the corresponding count is added by one, otherwise the current child node is added under the parent node. This builds the final SST tree through a number of DOM trees. Next is the process of analyzing and pruning the SST tree, the pruning is based on the weights of each node. The weight of a node consists of two main parts, the node's own weight plus the weight of all the node's children. The calculation is as in equation

(1):

$$NodeWeight(E)=\begin{cases}-\sum_{i=1}^{l} p_i \log_m p_i & if\ m>1\\ 1 & if\ m=1\end{cases} \tag{1}$$

where m denotes the number of web pages containing the node E and l denotes the number of child nodes contained at several points of E in the SST tree.

And the final weight of a node is calculated as in equation (2), where λ is a conditioning factor:

$$\begin{aligned} CompWeight(E) &= (1-\lambda^l)\, NodeWeight(E) \\ &\quad +\lambda^l \sum_{i=1}^{l} CompWeight(Si) \\ CompWeight(Si) &= \frac{\sum_{j=1}^{k} CompWeight(E_j)}{k} \end{aligned} \tag{2}$$

In the process of calculating the comprehensive weights of the nodes described above, it was found that there is sometimes a bias in the calculation of the weights of the nodes when using a simple arithmetic average of all nodes. The calculation of the combined weights is optimized to use the weighted average method, where P_{ij} denotes the frequency of the node E_j appearing in the node Si, as in Equation (3):

$$CompWeight(Si)=\frac{\sum_{j=1}^{k} P_{ij} * CompWeight(E_j)}{k} \tag{3}$$

2.2 Clustering of informational text

2.2.1 Density-based clustering algorithm

Density-based clustering algorithms are increasingly active in the research subfield of multidimensional spatial data mining, unlike other traditional clustering algorithms, the algorithm theoretically uses the number of elements in a unit region in space as a constraint on the algorithm's search to discover clusters of arbitrarily shaped element sets.

It is assumed that the elements are distributed in a certain space, and each element has its own fixed position in the space, and the distance that exists between each element. Whenever the density of an element's neighboring region exceeds a certain threshold, clustering is performed with this element at its core; in other words, a specific data point in a given class is tested to see if it contains at least a certain number of other element points in a region of a given radius. According to the classical algorithm of density-based clustering, such a clustering algorithm can ignore the interference “noise” data, and is suitable for discovering data clusters of arbitrary shapes.

The clustering process of the density-based clustering algorithm is shown in Figure 1.

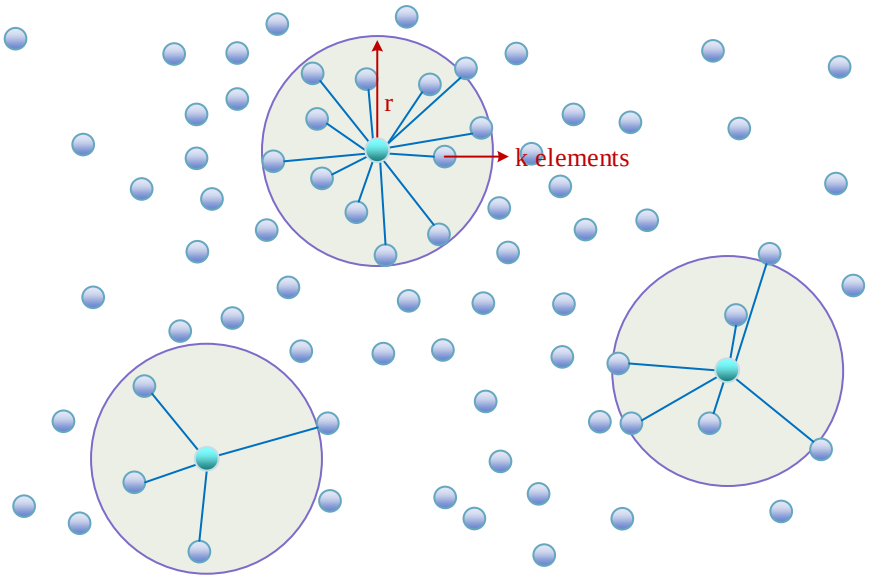


Figure 1: The description of the proximity radius r and the threshold k

Based on the above description, the key to clustering lies in the determination of the proximity region, the threshold value, and the position of the elements, which are three important factors in total.

Meanwhile, when designing the clustering algorithm, the following conditions should be satisfied as much as possible:

(1) Traversing the data only once, it can divide the large dataset into sub-parts and find the appropriate clustering core.

(2) Using limited memory resources and CPU resources, the processing is carried out in parts, and it is better to realize multi-threaded uncoupled processing.

(3) Although the proximity region is a circle with a fixed radius, the clustering clusters are of arbitrary shapes, so when designing the algorithm, we cannot simply search by radius.

(4) It can better resist the interference of noisy data.

(5) The clustering time is linear with the number of data objects and is independent of the data order.

(6) Suitable for clustering of large, high-dimensional data sets.

2.2.2 Data meta-clustering algorithms

According to the elaboration of the density-based clustering algorithm above, the specific steps for designing the data element clustering algorithm are as follows:

(1) Import the CEW mining results and perform disambiguation on the feature strings of the data elements.

(2) Use the designed filtering function to disambiguate the disambiguation results.

(3) Set the radius of the proximity region as r and the density threshold as k.

(4) Construct an undirected graph G, where the nodes of the graph G are the elements N_i in the set of data elements R_N, and the edges of the graph G are defined as follows: assume that the function D_{data} is the feature string correlation function, and let T_i be the feature string of N_i. If $D_{data}(T_j,T_k)>0$ and $D_{data}(T_j,T_k)<r$, then set an edge of length $D_{data}(T_j,T_k)$ exists between nodes N_j and N_k.

(5) Examine all the elements in the set R_N, and if its feature string contains consecutive

valid words (the result of CEW mining), then set the node as a starting point. Then there exists a subset R_S of the set R_N, and all elements in R_S are CEW words, i.e., all elements in R_S are starting points.

(6) Iterate over the elements n_{si} in R_S and check all nodes in R_N whose shortest distance from them on the graph is less than r (radius), which are considered to be related to the node n_{si}. If more than k (density threshold) related nodes are found, N_{si} is considered to be the core node of the public opinion event, and the CEW words contained in it are the core words of the public opinion event.

As can be seen from the above formulation, the clustering algorithm not only clusters the associated nodes together, but also finds the set of words that are likely to be the core words of the public opinion events.

The core difficulty of the data meta-clustering algorithm is how to construct a reasonable feature string association function D_{data}. The feature string is composed of several decomposition words, and thus the correlation degree of the inferred feature string is closely related to the correlation degree between the decomposition words that compose them. Before constructing D_{data}, let's construct the decomposition word association function D_{word}.

If two decomposition words often appear in the same feature string, then these two decomposition words are more closely related, accordingly set x as the number of times the two decomposition words w_1 and w_2 which are not the same appear simultaneously in all feature strings, then it can be inferred:

(1) x rises monotonically and $D_{word}(w_1, w_2, x)$ should fall monotonically and converge to 0 infinitely;

(2) x is a natural number;

(3) When $x = 0$, $D_{word}(w_1, w_2, x)$ should tend to infinity (with respect to r);

(4) When $x = 1$, $D_{word}(w_1, w_2, x)$ should approach r (either large or small).

In the real Internet environment, new articles are always released constantly, and the value of the correlation function between words based on feature strings is updated constantly, so if the relationship between two words is quantified purely from the number of times the two words appear in a feature string at the same time, the algorithm's ability to detect news events will inevitably be reduced with the passage of time, so this is not reasonable.

In this regard, the effect of time on the value of the correlation function is introduced, that is to say, the effect of the distance of the average publication time of the feature string (group) from the current time Δt on the relationship between the two words is taken into account.

Combining the above constraints on the correlation function, the correlation function is constructed as in equation (4):

$$D_{word}(x) = \frac{\theta r}{x^t + 0.0001} \tag{4}$$

where r is the radius of the proximity region, θ is the radius correction parameter, θ is calculated by taking into account the information outside the feature string, such as the length of the BBS body, the number of times it has been quoted, the combined influence of the posters and so on, in this paper, θ is taken purely based on the average influence of the posters of the data elements, and the influence is estimated according to the other topics in the The influence is estimated based on the leader influence model calculated in other topics. t is a time

correction parameter. It is set that the longer an element is published from the current time, the less effective it is in terms of public opinion.

After discussing $D_{word}(x)$, let's discuss the construction of the correlation function $D_{data}(t_1,t_2)$ for measuring the feature strings t_1,t_2 based on it.

The correlation between feature strings is determined by the correlation between the effective words that make up the strings, and at the same time, the more effective words the strings contain, the lower the weight averaged over each effective word. Based on this idea, the following feature string association function is constructed:

Let after filtering $t_1=\{w_{k_1},w_{k_2},\cdots\cdots w_{k_m}\}$, and $t_2=\{w_{l_1},w_{l_2},\cdots\cdots w_{l_n}\}$, X is the set of parameters x, then we have equation (5):

$$D_{data}(t_1,t_2,X)=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}D_{word}(w_i,w_j,x_{ij})}{m\times n} \tag{5}$$

Since $D_{word}(w_i,w_j)$ satisfies all the properties of the correlation function, and $D_{data}(t_1,t_2,X)$ is a linear summation of $D_{word}(w_i,w_j)$, and there is no constant term in this linear formula, it is deduced that the elements x_{ij} in the set of $D_{data}(t_1,t_2,X)$ pairs of parameters X also satisfy the following four conditions:

(1) $D_{data}(t_1,t_2,X)$ monotonically decreases when x_{ij} monotonically increases.

(2) $D_{data}(t_1,t_2,X)$ tends to infinity when $x_{ij}=0$ for any i,j.

(3) $D_{data}(t_1,t_3,X)\le D_{data}(t_1,t_2,X)+D_{data}(t_2,t_3,X)$.

From the previous definition of correlation function, $D_{data}(t_1,t_2,X)$ can be used as a correlation function to measure the correlation degree between feature strings T_1,T_2.

In this paper, the correlation degree of the feature strings is used to describe the correlation degree between the data elements, thus establishing the expression of $D_{data}(t_1,t_2,X)$, which also determines the method of constructing the undirected graph G used in the clustering algorithm.

3 Keyword-based generative summarization model for news

3.1 Calculation of keyword importance

This chapter calculates the importance of words based on the TextRank algorithm and fuses the semantic information and LDA topic modeling, which includes four modules: word representation model module, topic differentiation module, semantic relevance calculation module, and probability transfer matrix construction module, and its specific structure is shown in Figure 2.

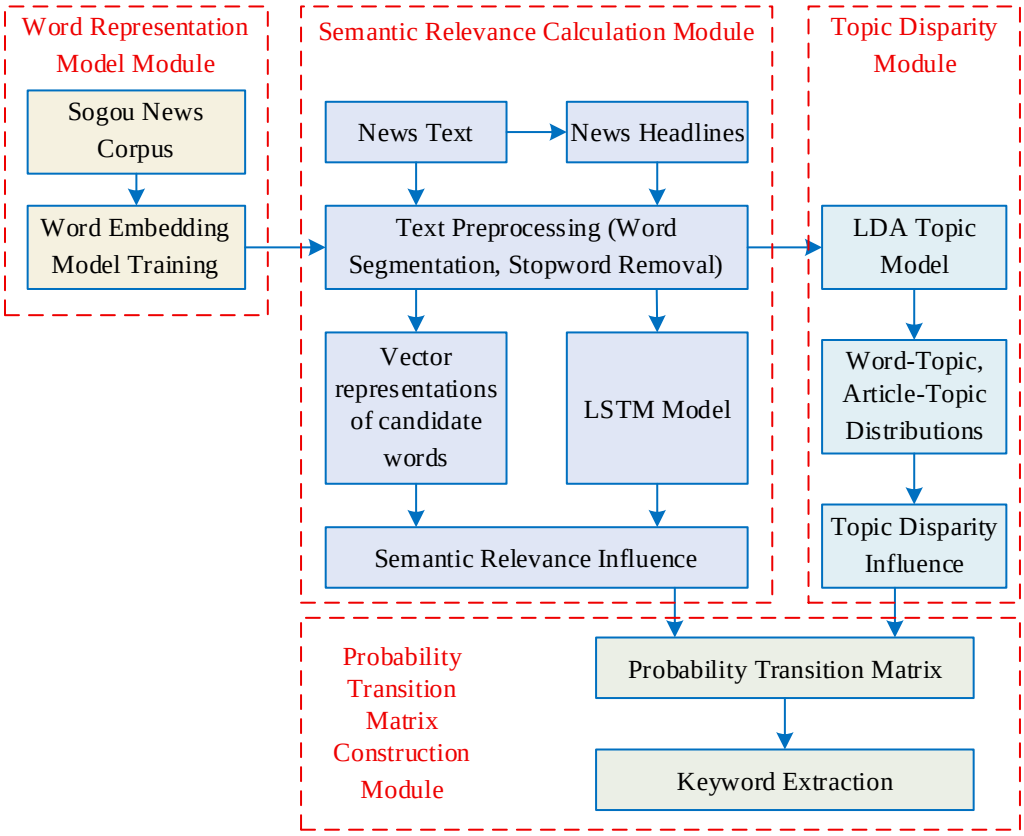


Figure 2: The structure of the calculation method for word importance

3.1.1 Word Representation Module

For the training of word representation model, this paper chooses Google's open-source toolkit Word2vec, and adopts Skip-gram model to train the word representation model on the whole network news corpus will be the window size of 5, and the vector dimension is 200 dimensions.

3.1.2 Thematic Differentiation Module

In order to characterize the difference and coverage between keywords, the topic difference impact degree is introduced to indicate the sharing rate of candidate keywords among different topics. The topic difference influence degree of candidate keywords is obtained by LDA topic modeling.

The structure of LDA topic model is shown in Fig. 3, which considers that a document consists of several topics, and each topic consists of several words. In the figure, M is the number of documents, T is the number of topics, N_m is the total number of feature words in the m th document, and $W_{m,n}$ and $Z_{m,n}$ are the n th feature word and its topic in the m th document, respectively. ϑ_m and ϕ_t are the Dirichlet prior distributions obeying the hyperparameters α and β, respectively, ϑ_m is the topic probability distribution of the m th document, and ϕ_t is the probability distribution of the feature words in topic t. The hyperparameters $\alpha = 0.1$ and $\beta = 0.1$ are set to obtain the probability $p(t \mid m)$ that the m th document belongs to topic t and the probability $p(t \mid u, m)$ that the word u in the m th document is generated by the topic t through the LDA topic model.

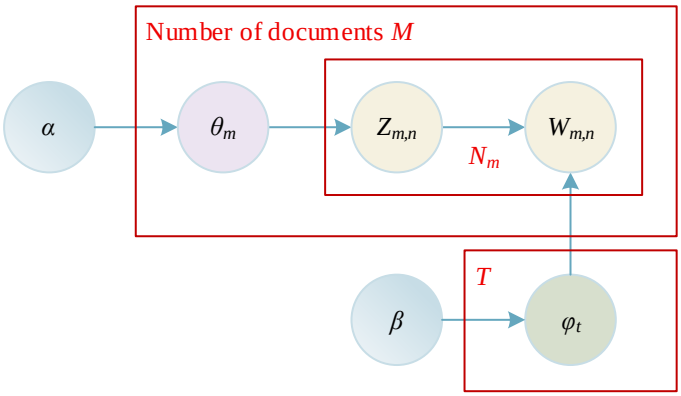


Figure 3: LDA Topic Model

Define $TS(u)$ as u 's Topic Difference Influence in a document, characterizing the variability and coverage among keywords, which is the sharing rate of candidate keywords among different topics. The Kullback-Leibler distance is used to measure the topic differentiation $TS(u)$, which is calculated as in equations (6)-(8):

$$TS(u) = \sum_{t=1}^{T} p(t \mid u, m) \log \frac{p(t \mid u, m)}{p(t \mid m)} \tag{6}$$

$$p(t \mid m) = \theta_t^m \tag{7}$$

$$p(t \mid u, m) = \frac{\phi_u^t \theta_t^m}{\sum_{z=1}^{T} \phi_u^z \theta_z^m} \tag{8}$$

T is the number of topics; θ_t^m is the probability of topic t in the m th document, and $p(t \mid m)$ is the probability that the m th document belongs to topic t; ϕ_u^t is the probability that the word u belongs to the topic t, and $p(t \mid u, m)$ is the probability that the word u is generated by the potential topic t in the m th document.

3.1.3 Semantic Relevance Calculation Module

The nouns, verbs and adjectives in the preprocessed news text are selected as candidate keywords. Combine the trained word2vec model to get the candidate word vector representation, use the LSTM model to get the news title vector, calculate the semantic relevance influence degree of the candidate keywords and the title, and construct the network model for calculating the semantic relevance is shown in Fig. 4.

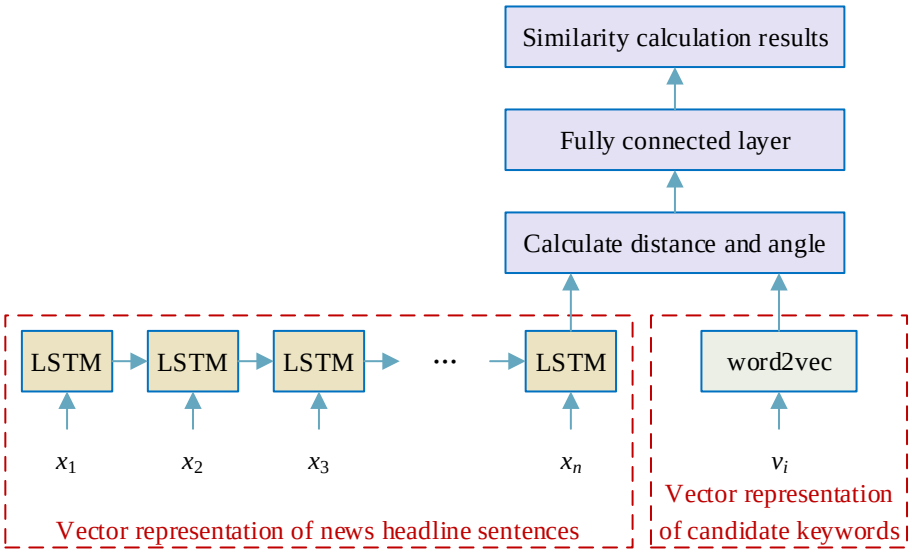


Figure 4: A network model of semantic relevance

3.1.4 News headline sentence vector representation

After text preprocessing of news headlines, we get a collection of news headline words $H=\{x_1,x_2,...,x_{n-1},x_n\}$, where n is the number of headline words; then the news headline word vector is initialized and sequentially input into the LSTM model to get the hidden state as in equation (9):

$$h_t = LSTM\left(h_{t-1}, x_t\right) \tag{9}$$

h_{t-1} is the hidden state of the last time step, $t\in[1,n]$; x_t is the input of the news headline phrase at the current moment, $t\in[1,n]$; sequentially, the news headline phrase is inputted into the LSTM model, and the hidden state of the last time step, h_n, is outputted to the model as a vector representation of the news headline sentence.

3.1.5 Semantic relevance calculation

The news headline sentence vector representation h_n obtained by using LSTM model and the candidate keyword vector representation v_i obtained by using word2vec model, and the distance and angle of the two are calculated as in Eqs. (10)-(11):

$$h_x^i = h_n \cdot v_i \tag{10}$$

$$h_+^i = \left|h_n - v_i\right| \tag{11}$$

h_x^i is the angular difference between the news headline sentence vector and the candidate keyword vector; h_+^i is the distance difference between the news headline sentence vector and the candidate keyword vector, and both of them are inputted into the fully connected layer as in Eqs. (12)-(13):

$$h_s^i = \sigma\left(w_1 h_x^i + w_2 h_+^i + b_1\right) \tag{12}$$

$$\tilde{p}_i = soft\max\left(w_3 h_s^i + b_2\right) \tag{13}$$

h_s^i is the output vector of the fully-connected layer, w_1, w_2, b_1 are the parameters of the fully-connected layer, which are learnable variables, and the parameters of the fully-connected layer are all matrices or vectors. The probability distribution $\tilde{p}_i$ is obtained by normalizing the output layer softmax function, w_3 and b_2 are the learnable parameters of the output layer, which are vectors, respectively. In turn, the similarity score y_i' of word-sentence pairs is obtained as in equation (14):

$$y_i' = r\tilde{p}_i^T \tag{14}$$

$r = [0, 0.5, 1]$, $\tilde{p}_i^T$ is the transpose of the probability distribution $\tilde{p}_i$. Since $y_{i'} = r\tilde{p}_i^T \approx y_i$, where y_i denotes the manual scoring. The manual scoring adopts the method of multiple cross-scoring, and takes the most rating levels obtained by each word-sentence pair as its similarity score. The target distribution is defined to satisfy $y_i = r\tilde{p}_i^T$ as in equation (15):

$$\tilde{p}_i^j = \begin{cases} 1-(y_i - 0.5*\{y_i\})/0.5 & if\ j = \lceil y_i + 0.5 \rceil - 1 \\ (y_i - 0.5*\{y_i\})/0.5 & if\ j = \lceil y_i + 0.5 \rceil \\ 0 & Other \end{cases} \tag{15}$$

$j \in [0, 2]$, is each dimension of the probability distribution $\tilde{p}_i$; $\lceil s \rceil$ is the smallest integer greater than s; and $\{s\}$ denotes rounding up when $s <= 0.5$, and rounding down to one place when $s > 0.5$.

The loss function is given by equation (16):

$$J(\theta) = \frac{1}{K}\sum_{i=1}^{K} KL\left(p(\theta)^i \,\|\, \tilde{p}(\theta)^i\right) \tag{16}$$

K is the total number of samples for training; i is the ith word-sentence pair sample. The semantic similarity between candidate keywords and news headline sentences is obtained by training the semantic relevance computation model.

3.1.6 Construction of the probability transfer matrix module

According to TextRank algorithm, given the jump probability transfer matrix between nodes in the word graph. Let the matrix M denote the probability transfer matrix as in equation (17):

$$M = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nn} \end{bmatrix} \tag{17}$$

The jth column in M denotes the probability distribution of jumping from word node j to other nodes, and the sum of the jump probabilities in each column is 1.

On the basis of the coverage influence of the traditional TextRank algorithm, we add the topic differentiation influence and semantic relevance influence, so that $p_f(v \to a)$ denotes the transfer probability of the coverage influence, i.e., Eq. (18):

$$p_f(v \to a) = \frac{1}{E(v)} \tag{18}$$

E is the set of neighboring nodes of node v. Let $p_t(v \to a)$ denote the transfer probability of thematic differential influence, calculated as in equation (19):

$$p_t(v \to a) = \frac{TS(a)}{\sum_{u \in adj(v)} TS(u)} \tag{19}$$

$TS(a)$ is the thematic distinctiveness influence degree of the word a, and $adj(v)$ is the set of neighboring nodes of v.

Let $p_y(v \to a)$ denote the transfer probability of the semantic relevance influence degree, computed as in equation (20):

$$p_y(v \to a) = \frac{y'(a)}{\sum_{u \in adj(v)} y'(u)} \tag{20}$$

$p_y(v \to a)$ is the transfer probability of semantic relevance influence degree from node v to node a. $y'(a)$ is the semantic relevance influence degree of the word a and $adj(v)$ is the set of neighboring nodes of v.

The three transfer probabilities are fused to obtain the final probabilistic transfer probability $p(v \to a)$ as in equation (21):

$$p(v \to a) = \phi p_f(v \to a) + \varphi p_t(v \to a) + \gamma p_y(v \to a) \tag{21}$$

where ϕ, φ, γ are the weights of the three transfer probabilities, $\phi + \varphi + \gamma = 1$, respectively.

The final probability transfer matrix M is formed by $p(v \to a)$.

3.1.7 Keyword extraction

During the iterative computation of extracted keywords, assuming that there are n candidate keywords in the text, the importance scores of all nodes are homogenized to obtain the initial importance score vector B_0 as in equation (22):

$$B_0 = \left[\frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \cdots, \frac{1}{n}\right] \tag{22}$$

The final transfer matrix constructed using the fusion coverage influence, thematic difference influence, and semantic relevance influence is iteratively calculated as in equation (23):

$$B_i = d \times M \times B_{i-1} + (1-d) \times \frac{e}{n} \tag{23}$$

B_i is a vector of node importance scores for the current iteration; d is the damping coefficient; n is the total number of nodes; and e is an n-dimensional vector with all components 1. The iteration is considered converged when the difference between the results of two iterations B_i and B_{i-1} is less than a set threshold. The vector B_i is considered to be the final importance score of the node, which is sorted in descending order, and the topK words are selected as keywords.

3.2 A generative summarization model that incorporates structural and contextual information

3.2.1 Overview of the model

Based on the generative summarization paradigm of multi-source information fusion, a generative code summarization model (CodeSum) is constructed based on keyword-guided fusion of structural and contextual information, which utilizes structural and contextual information of the source code to enhance the encoder's ability to represent the code, and utilizes the keyword information to guide the decoder to generate summaries that are faithful to the original topic. The whole contains a multi-component encoder, decoder and keyword guidance module. The sequence encoder is first utilized to encode the content of the code itself. A dual graph structure encoder is used to learn the global and local structure information of the code, and a context encoder is used to encode the external context information. Then keyword information is extracted and encoded from the source code to guide summary generation. Finally, a dynamic word list strategy is performed.

3.2.2 Multi-component encoder

(1) Code Content Encoder

For the code fragment sequence $w_1, w_2, \ldots, w_n$, it is initialized as a vector sequence $e^c = \{e_1^c, e_2^c, \ldots, e_n^c\}$ through the word embedding layer, and encoded as a hidden vector sequence $h^c = \{h_1^c, h_2^c, \ldots, h_n^c\}$ using the GRU layer to compute the representation of each token, h_i^c is computed as in equation (24):

$$h_i^c = GRU\left(h_{i-1}^c, e_i^c\right) \tag{24}$$

(2) Code structure encoder

The AST-based dual graph encoder is used in the model, and its structure is shown in Fig. 5. Its top-down transmission allows the child nodes in the AST tree to obtain the information of their parent nodes, and after many information transfers, the local information of each leaf node will be more aggregated and enriched. Whereas bottom-up transmission allows the parent node to obtain information about its child nodes, and after multiple information transfers, the root node will aggregate the global information of the entire code structure.

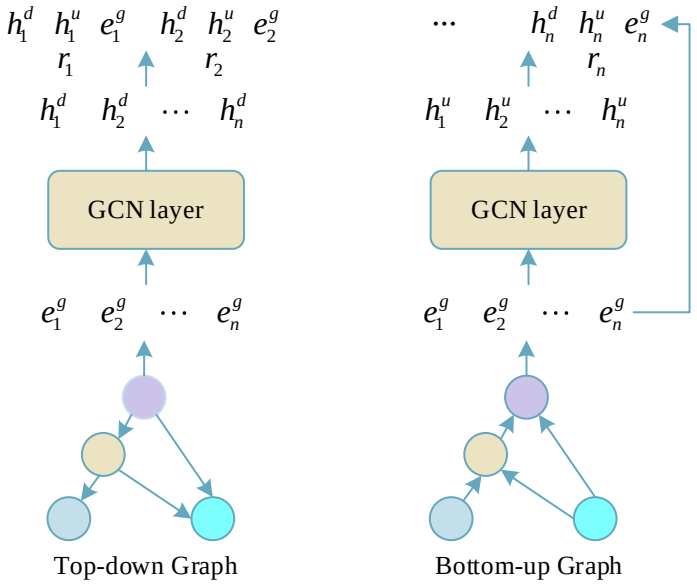


Figure 5: Code structure encoder

Represent the AST of code fragments as top-down and bottom-up graphs. Encode the nodes in the graphs as a sequence of vectors $e^g = \{e_1^g, e_2^g, \ldots, e_n^g\}$ via a shared word embedding layer. The node vector sequences e^g and and their edge relations of the two graphs are input into two independent graph convolutional neural networks to obtain the updated node vector sequences h^d and h^u as in Eqs. (25)-(26):

$$h_i^d = GCN_d\left(\left\{e_i^g, e_j^g : j \in edge_d(i)\right\}\right) \tag{25}$$

$$h_i^u = GCN_u\left(\left\{e_i^g, e_j^g : j \in edge_u(i)\right\}\right) \tag{26}$$

$edge_d(i)$ and $edge_u(i)$ are the edge relations of node i in the top-down and bottom-up graphs, respectively.

Combining the node vectors h_i^d and h_i^u and the node vector e_i^g of the top-down and bottom-up graphs, the node representation r_i with more complete structural information is obtained as in Equation (27):

$$r_i = \left[h_i^d \left\| h_i^u \right\| e_i^g\right] \tag{27}$$

Since the node representation r_i combines graph information from two different directions, it is encoded using a bidirectional GRU as in Eqs. (28)-(30):

$$\overrightarrow{h_i^g} = GRU_f\left(r_i, \overrightarrow{h_{i-1}^g}\right) \tag{28}$$

$$\overleftarrow{h_i^g} = GRU_b\left(r_i, \overleftarrow{h_{i-1}^g}\right) \tag{29}$$

$$h_i^g = \left[\overrightarrow{h_i^g} \left\| \overleftarrow{h_i^g} \right. \right] \tag{30}$$

Finally, the code structure encoder generates a set of node representations h^g containing information about the local and global structure of the code.

(3) Code Context Encoder

The model also integrates a code context encoder to enhance the contextual semantic information of the code, using relevant contextual code snippets from the same file as the contextual information of the input code snippets. The maximum number of context code snippets per input code fragment is set to m and the maximum number of tokens per context code snippet is set to n. Convert the sequence of context code snippet f_i code text into a sequence of word vectors e^{f_i} via the word embedding layer. The output vector h^{f_i} of the GRU layer is directly obtained as the vector representation of the i th context code fragment, and the vector representations of all the context code fragments are integrated as the code context information representation, notated as $h^f = \left[h^{f_1}, h^{f_2}, cdots, h^{f_m} \right]$.

(4) Fusion multi-component encoder

The multi-attention mechanism is utilized to integrate the above three encoders in a unified framework, where the hidden vector sequence of each component encoder is compressed into a single vector representation, which is computed as in Eqs. (31)-(32) by the sequence of encoder hidden vectors, h^*, and the sequence of decoder hidden vectors, s:

$$v^* = h^* s^T \tag{31}$$

$$a^* = soft\max\left(v^*\right) \tag{32}$$

The encoder information is compressed into a context vector c^* through the attention distribution a^* as in equation (33):

$$c^* = a^* h^* \tag{33}$$

Substituting h^c, h^g and h^f into the corresponding formulas to obtain c^c, c^g and c^f, respectively, yields the fused of the encoder context vector c as in equation (34):

$$c = \left[c^c \left\| c^g \right\| c^f \right] \tag{34}$$

3.2.3 Decoders

In the decoding stage, a partially generated summary $y = [y_1, y_2, \ldots, y_{t-1}]$ is used as input to the decoder, and a word embedding layer is used to convert the summary words into a sequence of word vectors $e^s = [e_1^s, e_2^s, \ldots, e_{t-1}^s]$. The sequence of word vectors e^s is fed into the GRU layer to generate a sequence of decoder hidden layer vectors s. The last hidden layer vector h_n^c of the code content encoder is passed into the GRU layer as the initial hidden layer state. The word probability distribution for the current time step is computed Eq. (35):

$$P_{vocab}\left(y_t \left| y_1, y_2, \ldots, y_{t-1}, c\right.\right) = soft\max\left(dense\left([c, s]\right)\right) \tag{35}$$

P_{vocab} is the word probability distribution for all words in the word list, which provides the final probability distribution for predicting the next word w as in equation (36):

$$P(w) = P_{vocab}(w) \tag{36}$$

3.2.4 Keyword Guidance Module

Construct a keyword guidance module to avoid the model to be disturbed and deviate from the key topic information. Extract token from function name as keyword. Encode the keyword as a sequence of word vectors e^k using the word embedding layer, and capture the semantics of the keyword using the GRU layer to obtain a hidden vector sequence representation of the keyword $h^k = [h_1^k, h_2^k, \ldots, h_l^k]$.

A keyword-based replication mechanism is used to guide the generation of code summaries, and the generation probability p_{gen} is calculated by equation (37):

$$p_{gen} = sigmoid\left(\frac{1}{n_s}\sum W_p[c^k, s, e^s] + b_p\right) \tag{37}$$

W_p and b_p are learnable parameters and n_s is the length of the decoder input sequence. p_{gen} is the probability of generating a word from the word list, while $1 - p_{gen}$ is the probability of copying a word from the original input text. The final word probability distribution $P(w)$ is obtained through equation (38):

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen})\sum_{i:w_i} a_i^k \tag{38}$$

a_i^k is the attention distribution of the keywords. $P(w)$ represents the probability of word w, thus the word with the highest probability in the word list is the summary word generated at the current time step. The goal of model training is to minimize the loss function of Eq. (39):

$$\ell(\theta) = -\frac{1}{L}\sum_{t=1}^{L} \log P\left(y_t \middle| c, y_1, y_2, \ldots, y_{t-1}\right) \tag{39}$$

L is the length of the target summary.

3.2.5 Dynamic word strategies

A dynamic word list strategy is used for automatic code summarization. This section aims to improve the quality of generated summaries while saving time and space costs, and the key to improving the efficiency of the model run is how to reduce the size of the word list. Since a particularly large word list is not required during model operation, this paper constructs a dynamic word list for each batch of data. The reconstructed dynamic word list consists of two parts: data word list and high-frequency word list. The data word list includes the word V_x^c in the input code and the word V_x^s in the reference summary. And the high-frequency word list V_x^h comes from the words that appear more frequently in the whole dataset. In this paper, the words in the reference summaries only exist in the model training and validation phases, and

are not considered in the testing phase. A dynamic word list V_x is constructed for each input code fragment x in the model training and validation phase as in equation (40):

$$V_x = V_x^h \cup V_x^c \cup V_x^s \tag{40}$$

The dynamic word list V_x constructed for each input code fragment x in the model testing phase is equation (41):

$$V_x = V_x^h \cup V_x^c \tag{41}$$

V_x^h is the high frequency word list, V_x^c is the source code data word list, and V_x^s is the reference summary data word list. The word list of a batch is the concatenation of the word list V_x of all data in a batch as in equation (42):

$$V_{batch} = V_{x_1} \cup V_{x_2} \cup \ldots \cup V_{x_b} \tag{42}$$

b is the size of a batch.

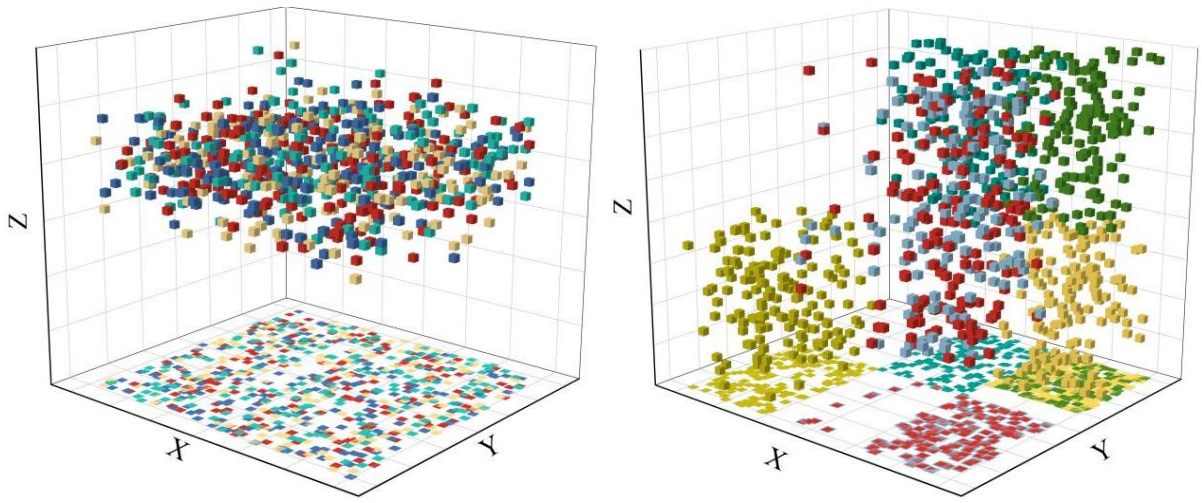
4 Integrating artificial intelligence and data mining for news production testing

4.1 Validation of the effectiveness of the density-based data meta-clustering algorithm

4.1.1 Visualization of news information clustering

In order to better validate the density-based news information clustering algorithm (data metaclustering algorithm), the embedding of its two important modules is compared and tested to verify the effectiveness of this clustering framework. From the second chapter, it is known that the core of the density-based data metaclustering algorithm lies in the construction of the decomposition word association function and the calculation of the feature string correlation degree, so the two are divided into modules for the comparison of the effectiveness of the module ablation experiments.

The clustering without constructing the decomposition word association function is shown in Fig. 6(a), and its clustering effect is poor, there is no boundary between the clusters and clusters, and the clusters are not only fused together but also divided into only 4 classes. The clustering without considering the average publication time of the feature string from the current time distance is shown in Fig. 6(b), its clustering effect is slightly stronger, the clusters and clusters have a slightly clear division of the boundary between the clusters and clusters have 5 types of clusters. The complete clustering algorithm running effect is shown in Figure 7, after clustering the clusters and clusters between the division and boundary is more obvious, the number of clusters up to 8, clustering effect is better.



(a) (b)

Figure 6: Clustering performance after module ablation

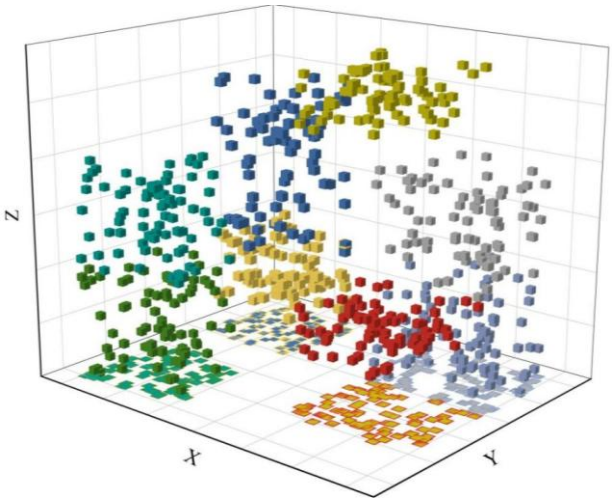


Figure 7: The clustering performance of the complete algorithm

4.1.2 Clustering effects

Take (B1) K-Means algorithm and (B2) LF algorithm as the experimental comparison algorithm, to carry out with (B3) this paper algorithm on the random 10 different topics of news information clustering on the checking accuracy, checking all the rate and the F1 value of the performance comparison is shown in Table 1. News events are numbered using the numbers 1-10 and cover the following topics: celebrity controversies, national territorial conflicts, social oddities, child safety, food safety, aviation processes, sports events, international political changes, science and technology, and art exhibitions. The performance of the three indicators of the three algorithms are all above 0.80, indicating that they all have a certain clustering effect. From a single indicator, (B3) this paper's algorithm is the best performance among the three algorithms, and its clustering effect on 10 news events with different subject matter content has a check accuracy rate, a check completeness rate and an F1 value of >0.900, of which the average check accuracy rate is 0.91, the average check completeness rate is 0.92, and the average F1 value is 0.93.

Table 1: Comparison of evaluation indicators of the three algorithms

	Precision rate			Recall rate			F1 value		
	B1	B2	B3	B1	B2	B3	B1	B2	B3
1	0.91	0.94	0.91	0.86	0.87	0.97	0.83	0.9	0.99
2	0.86	0.86	0.88	0.88	0.88	0.89	0.94	0.94	0.95
3	0.87	0.86	0.89	0.93	0.94	0.9	0.8	0.92	0.93
4	0.95	0.83	0.98	0.84	0.91	0.88	0.95	0.83	0.86
5	0.95	0.84	0.86	0.84	0.81	1.00	0.88	0.82	0.87
6	0.90	0.81	0.88	0.84	0.88	0.9	0.9	0.89	0.99
7	0.93	0.81	0.88	0.85	0.85	0.93	0.93	0.91	0.89
8	0.92	0.89	0.87	0.83	0.86	0.95	0.9	0.88	0.95
9	0.86	0.95	0.99	0.8	0.94	0.94	0.93	0.93	0.96
10	0.86	0.85	1.00	0.92	0.83	0.88	0.92	0.85	0.86
Average	0.90	0.86	0.91	0.86	0.88	0.92	0.90	0.89	0.93

Combing the 3 algorithms in the implementation of the events in the clustering of the 10 themes is shown in Table 2, the different algorithms for the clustering of news events of different themes content implementation time varies greatly, but the overall implementation time range in the (10.00,90.00)s interval, with a certain clustering implementation speed. And among the three algorithms, the execution efficiency of (B3) this paper's algorithm is the highest, and its average execution time is 34.19s, which is much smaller than (B1) K-Means algorithm (50.175s) and (B2) LF algorithm (52.967s).

Table 2: The execution time of algorithmic clustering

	Execution time (s)		
	B1	B2	B3
1	23.07	83.38	24.1
2	47.94	50.48	40.25
3	69.47	49.71	30.74
4	25.68	56.84	33.88
5	36.86	63.19	34.9
6	30.6	29.93	23.13
7	72.3	80.41	54.19
8	41.03	21.75	12.73
9	87.72	59.16	57.06
10	67.08	34.82	30.92
Average	50.175	52.967	34.19

4.2 Operational evaluation of the news generative summarization model

4.2.1 Calculation of word importance

A news event on the progress of the lunar exploration project is selected as an experimental sample, and the following words are used to categorize the content of the news event: earth-moon/space/abundance/minerals/water-ice/lunar soil/resources/becoming/new target. Using this paper's keyword importance calculation method, the webpage news is modeled, analyzed and calculated as part of the results of the descending order of the composite value is shown in Table 3. In the table, there are five keywords, namely, Earth-Moon space, lunar mining, lunar

surface engineering, resource utilization and lunar exploration, with composite values of 0.2000 or above, and the results of the processing indicate that these nodes as keywords are basically able to reflect the main idea of the original news, which verifies that the composite value of the keywords is 0.2000 or above. This verifies the practicality of the word importance calculation method under the influence of thematic difference and semantic relevance.

Table 3: Keyword ranking result

Sort	Node ID	Node character	Synthetical value
1	87	Lunar-earth space	0.5702
2	125	Lunar mining	0.4693
3	133	Lunar surface engineering	0.3618
4	74	Resource utilization	0.3473
5	118	Lunar exploration	0.2814
6	86	Space transportation	0.1954
7	111	Scientific experiments	0.1736
8	107	Resource exploration	0.1696
9	147	Space exploration	0.1618
10	45	Development strategy	0.1542
11	86	Llunar research station	0.1365
12	81	Lunar base	0.1305
13	79	Industrialization	0.1213
14	99	Intelligent production	0.1149
15	102	Space Mining	0.1081

4.2.2 Accuracy of Generated Summaries with Multiple Attention Mechanisms

Using the news topic text of "Exploitation and Utilization of Rich Mineral, Water Ice and Lunar Soil Resources in the Earth-Moon Space" as the experimental material, (C1) uses a generative summary model without the multi-attention mechanism, (C2) replaces the multi-attention mechanism with the basic attention mechanism in the generative summary model, (C3) replaces the multi-attention mechanism with the self-attention mechanism in the generative summary model, (C4) the complete generative summary model serves as the comparison subject, and (C5) the reference summary is used as the comparison. The new word rate of the generated summaries of different subjects for 1 to 4 phrases and sentences under this news topic is shown in Figure 8. Although the overall trend of the new word rate of the three subjects is consistent with the performance of the reference abstract of (C4), all of them increase with the expansion of the phrase structure, the overall new word rate situation still has a big difference, in terms of the value of the high and low performance: (C1)(11.06%~61.30%) < (C2)(14.45%~85.88%) < (C3)(16.71%~89.71%) < (C4)(18.97%~ 91.01%), which verifies the effectiveness of the multi-attention mechanism in generating abstract abstracts with accuracy.

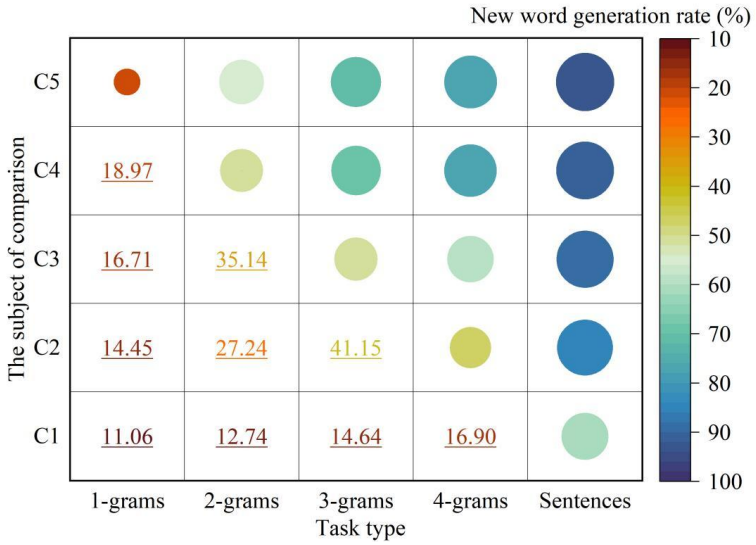


Figure 8: Comparison of the generation rate of new words for different themes

It is randomly inputted 5 news texts with different subject contents (numbered 1~5 in order), each of which contains 30 participles (represented by the numbers 1~30). The attention distribution of the output model to the five input text information is shown in Fig. 9. The color depth in the figure indicates the size of the probability distribution, the lighter the color indicates the greater the degree of attention to the input words, and this paper sets the interval from low to high to be (0,10). The X-axis indicates the participle within a single text, and the Y-axis indicates the news text of different subject contents.

From the figure, it can be seen that the multi-attention of this paper's model mainly focuses on the words in the first part of the input text, and the color of the front part is dark (>6.0). This is due to the specificity of the news text, which generally puts important information in the first paragraph. The multi-attention mechanism enhances the semantic features of the text and enables the decoder to pay more attention to the key information of the input text.

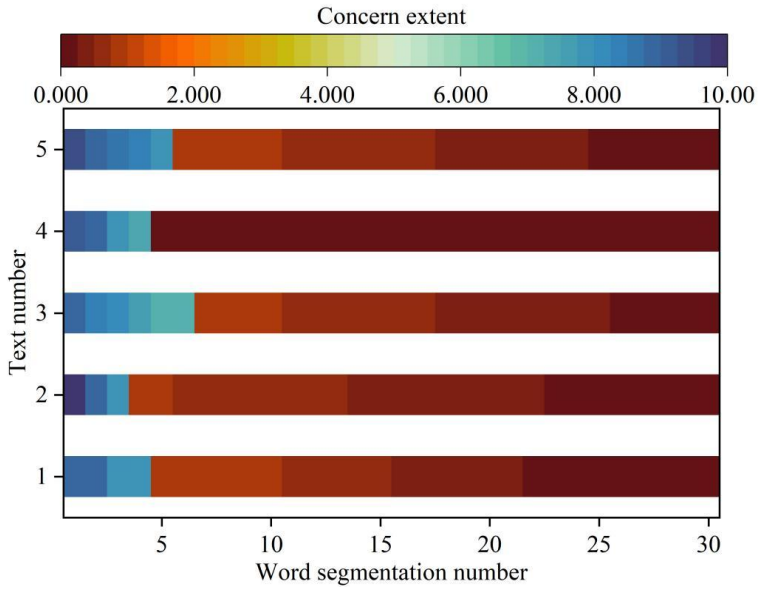


Figure 9: Visualization effect of the attention model

4.2.3 Level of generation of news summaries

The similarity assessment between the summaries decoded by the model and the reference summaries of the dataset is carried out with the ROUGE assessment mechanism, which is based on the principle that the similarity of two sentences is determined by the number of identical n-means in the sequence, and the higher the number of identical n-means, the higher the ROUGE score of the summaries decoded by the model. In this subsection, ROUGE-1, ROUGE-2, and ROUGE-L are used as the main evaluation indexes, and six commonly used models in the same field, namely NLP@WUST, NLP_ONE, AS-TKF, TIF-SR, SAKGT, and Pointer-generator+Coverage, are selected to statistically compare the experimental evaluations of the seven models. In the ROUGE-1 and ROUGE-L indicators, there are two models, ROUGE-1 and ROUGE-2, which do not participate in the evaluation, and all of them achieve the optimal performance of this paper's model (46.36% and 41.49%). On the ROUGE-2 indicator, the performance of the seven models is slightly weak, none of them exceeds 30.00%, but still the model of this paper is the best (29.78%). It shows that the news generative summarization model guided by the importance of keywords can extract more effective textual features from the textual news sequences, thus generating the semantics of news summaries with high accuracy, which verifies the high reliability of the generative summarization model fusing structural and contextual information in the task of news summarization generation.

Table 4: The evaluation results of the comparative experiments of seven models

Model	ROUGE-1	ROUGE-2	ROUGE-L
NLP@WUST	-	23.42	-
NLP_ONE	-	23.78	-
AS-TKF	38.49	24.89	33.66
TIF-SR	39.71	25.43	34.58
SAKGT	40.48	25.24	35.45
Pointer-generator+Coverage	44.55	27.92	39.34
Textual	46.36	29.78	41.49

5 Conclusion

The data metaclustering algorithm proposed in this paper has a better performance of clustering effect, not only the clustering cluster boundary division is obvious and detailed, the average checking accuracy rate is 0.91, the average checking completeness rate is 0.92, the average F1 value is 0.93, and the average execution time is only 31.19s for multiple topic content news time clustering.

The word importance calculation scheme designed based on the different attributes of semantic importance and coverage variability of key words is used to obtain the keyword importance by vectorizing the word representation and synthesizing the influence of thematic variance and the influence of semantic relevance. With this keyword-guided news generative summarization model, the new word generation rate for experimental tasks with different structures ranges from 18.97% to 91.01%, and the ROUGE-1, ROUGE-2, and ROUGE-L performances of the generated summaries are 46.36%, 29.78%, and 41.49% in turn, which fit the textual characteristics of the news topics, and have both superior generation accuracy and generation Quality.

Through the effective mining and accurate extraction of big data technology, the artificial intelligence technology under the fusion of structural and contextual information can realize the automatic refining and summarizing of news text content, assisting in the rapid and effective

acquisition of key information in the process of news production, and it is an important technology to alleviate the information overload and improve the efficiency of information acquisition.

About the Author

Ning Shuo was born in Huhehot, Inner Mongolia, China, in 1990. My main research direction is new media and society.

References

[1] Saleh, K. S., & Hassan, H. D. (2022). TV News Production In Smart Newsrooms Using Modern Technologies. Journal of Positive School Psychology, 6(5).

[2] Goyanes, M., & de Zúñiga, H. G. (2021). Citizen news content creation: Perceptions about professional journalists and the additive double moderating role of social and traditional media. Profesional de la información, 30(1).

[3] Farid, A. S. (2023). Changing the paradigm of traditional journalism to digital journalism: Impact on professionalism and journalism credibility. Journal International Dakwah and Communication, 3(1), 22-32.

[4] Al-Essa, M. R. K., & Hassouni, M. S. (2025). The Impact of New Media on Traditional Journalism. Central Asian Journal of Social Sciences and History, 6(4), 328-339.

[5] Janáčková, L. F. Ľ. (2024). News at the speed of AI: Automating journalism through text generator. MARKETING IDENTITY, 166.

[6] Tsourma, M., Zamichos, A., Efthymiadis, E., Drosou, A., & Tzovaras, D. (2021). An AI-enabled framework for real-time generation of news articles based on big EO data for disaster reporting. Future Internet, 13(6), 161.

[7] Pathak, A., Pawar, T., & Pawar, Y. (2025). SmartNews: AI-Powered News Summarizer. International Journal on Advanced Computer Theory and Engineering, 14(1), 475-481.

[8] Guo, Y., Wang, J., & Zhang, X. (2022, October). News Production and Business Communication Model of Internet Media Based on Big Data Technology. In 2022 World Automation Congress (WAC) (pp. 389-393). IEEE.

[9] He, Y. (2021). News Industry under the Background of Big Data. In E3S Web of Conferences (Vol. 275, p. 03050). EDP Sciences.

[10] Ai, Z., & Li, A. (2025). A Big Data-Driven Hybrid Recommendation Method for Accurate News Media Content Personalization. Journal of Circuits, Systems and Computers, 2650029.

[11] Altheneyan, A., & Alhadlaq, A. (2023). Big data ML-based fake news detection using distributed learning. IEEE Access, 11, 29447-29463.

[12] O'Halloran, K. L., Pal, G., & Jin, M. (2021). Multimodal approach to analysing big social

and news media data. Discourse, Context & Media, 40, 100467.

[13] Heim, S., & Chan-Olmsted, S. (2023). Consumer trust in AI–human news collaborative continuum: Preferences and influencing factors by news production phases. Journalism and Media, 4(3), 946-965.

[14] Ojoajogwu, H. M., Ter Akase, M., & Igyuve, A. I. (2025). Adoption of Artificial Intelligence in News Production by Select Broadcast Stations in North-Central, Nigeria. International Journal of Sub-Saharan African Research, 3(1), 111-124.

[15] Pleios, G., & Tastsoglou, M. (2025). AI and the News: Challenges Arisen From the Adoption of AI in News Production. Postmodernism Problems, 15(1), 3-24.