Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 5, 2144-2156
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i5.7419

News recommendation model and application in the ERA of integrated media based on knowledge graph

Shuhang Lyu[1*], Xueling Guan[2]
[1]Universiti Tunku Abdul Rahman (Sungai Long Campus) Negeri Selangor 47300, Malaysia; shuhanglyu@163.com (S.L.)
[2]Beijing Daxing Branch Shanghai Pudong Development Bank Beijing 102628, China; 2391718940@qq.com (X.G.).

Abstract: In the era of integrated media, the integration of news information has demonstrated an explosive proliferation. In this era, users cannot quickly and accurately find news of interest, and the importance of personalized news recommendations has become increasingly prominent. The model based on knowledge graphs can capture rich semantic and structural information and analyze users' personalized preferences more accurately. It is becoming the latest technology in recommendation systems. Based on the knowledge of knowledge graphs, this paper first studies the construction of knowledge graphs, including natural language processing, distributed computing, knowledge fusion, knowledge computing, and entity alignment. Then it researches the construction of mathematical models, including extracting word vectors, calculating user interest preference features, generating news feature vectors, convolutional neural network recommendation models, and integrating attention propagation layers. Finally, this paper designs and completes the simulation experiment. Using the Adressa data set, through the two evaluation indexes of F1 and AUC, the ANCNN model constructed in this paper is compared with the four benchmark models of DeepFM, DKN, NAML, and NPA, and the advantages of the ANCNN model constructed in this paper are verified. The research results have improved the efficiency of news selection and the reading experience.

Keywords: *Attention mechanism, CNN, Era of integrated media, Knowledge graph, News, Recommendation model.*

1. Introduction

Integrated media includes not only the integration of traditional media and emerging media, but also the integration of media content, platform, technology and channels, aiming at improving media communication effect and optimizing resource allocation. In the era of integrated media, news communication has broken through the limitations of time and space, and can be widely disseminated in an instant. The content of communication is extremely rich, the mode of communication is more diverse, the subject of communication is more extensive, and the interaction of communication is stronger [1, 2]. News information shows an explosive growth trend. In the face of a large number of news, users often feel at a loss when choosing, and they cannot quickly and accurately find news of interest. The importance of personalized news recommendation has become increasingly prominent. News recommendation can help users filter from numerous news resources and push news that may be of interest. At present, many news apps have embedded recommendation algorithms in their business systems to provide personalized news recommendations and enhance the user experience. For example, Toutiao constructs a news recommendation algorithm from three dimensions: news content characteristics, user personal characteristics and environmental characteristics, and successfully applies it to its platform. Compared with the recommendation of goods, movies and music, news recommendation has the characteristics of strong timeliness, rich semantic information and diverse user interests.

History: Received: 14 February 2025; Revised: 25 April 2025; Accepted: 29 April 2025; Published: 21 May 2025
* Correspondence: shuhanglyu@163.com

News recommendation has become a hot topic in the research of many scientific research institutions at home and abroad, which has attracted the attention of experts in the fields of information retrieval, data mining and artificial intelligence. As a structured and semantic knowledge representation method, knowledge graph is a graph database that stores massive knowledge in the real world. It has the characteristics of large scale, rich semantics, excellent quality and friendly structure [3]. It uses natural language processing technology to extract and construct related entities, enriches the overall analysis performance of the recommendation model, provides data support for a large number of knowledge-driven downstream tasks, and provides more abundant auxiliary information for personalized recommendation [4, 5]. Knowledge graph is becoming the latest technology of recommendation system. Using the advantages of deep learning technology in big data processing and feature extraction [6] auxiliary information to give full play to the advantages of neural network's powerful ability to learn graph structure. It excavates the connection between news information and the potential interest of readers, and reduces the time consumption in the recommendation process. And it can recommend more diversified news information resources according to the reader’s style or specific needs, effectively alleviate the problem of information overload, and improve the utilization rate of news information and the user's reading experience.

1.1. Construction of Knowledge Graph

The construction of KG is the basic work of this research. Its construction process is very complicated, but the core work of this topic is knowledge extraction and knowledge storage. The knowledge graph construction process is shown in Figure 1.

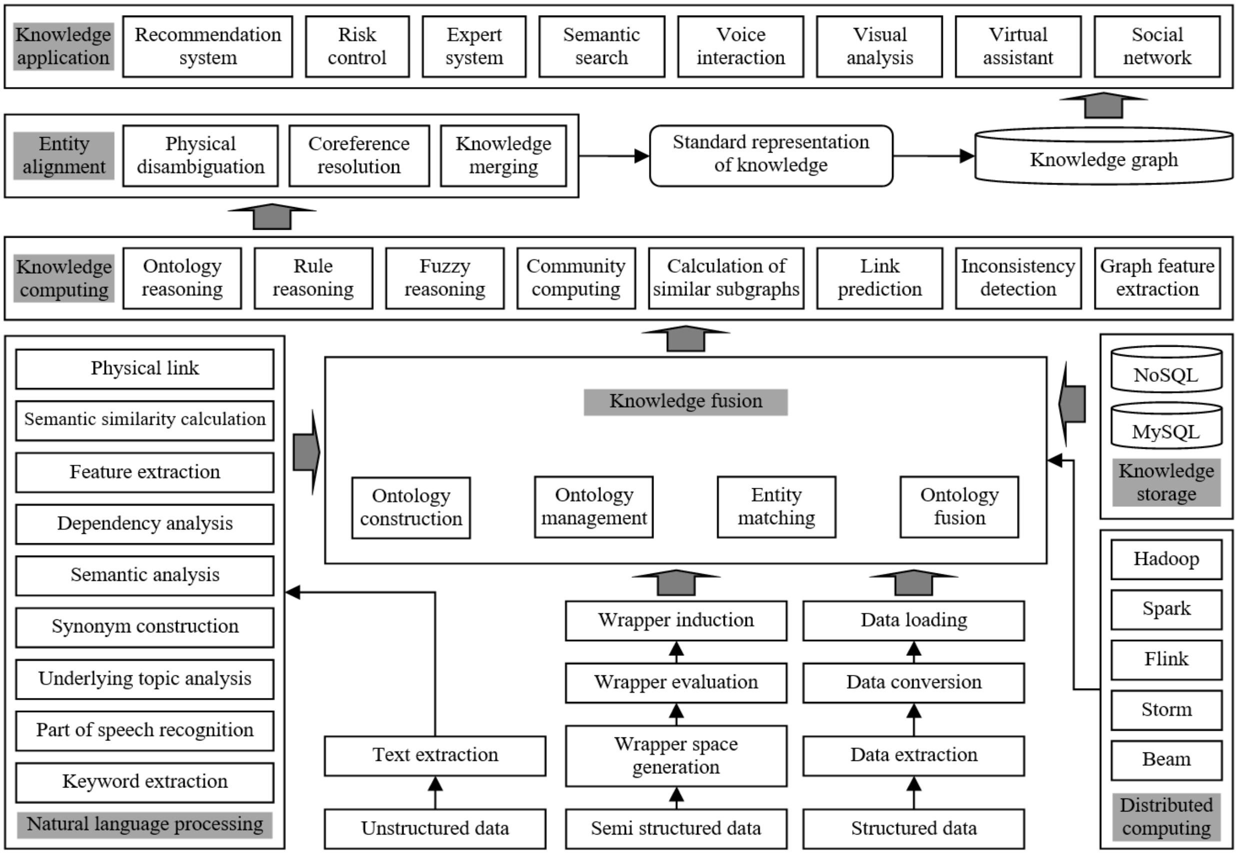

Figure 1.
Knowledge graph construction process.

1.2. Natural Language Processing

Natural language processing studies how to make computers understand, generate and translate human languages. The integration of natural language processing and knowledge graph can provide richer semantic information for natural language processing and provide more corpus data sources for knowledge graph. In addition, this combination can also provide more powerful understanding for artificial intelligence systems, thereby better supporting human-computer interaction. Firstly, the entities in the text are extracted and mapped to the corresponding entity nodes in the knowledge graph. Then, the relationship between entities is extracted from the text and mapped to the corresponding relationship edge in the knowledge graph. Finally, the entities and relationships in the knowledge graph are used to derive new knowledge.

1.2.1. Distributed Computing

There is an interdependent relationship between knowledge graph and distributed computing. As the scale of knowledge graph data continues to expand, the challenges of processing and storing knowledge graph data are also increasing. Therefore, distributed computing has become one of the key technologies of knowledge graph to solve the problem of large-scale data processing in knowledge graph [7]. Data fragmentation can divide large-scale knowledge graph data into smaller data blocks, thus solving the problem of processing large-scale data. Task allocation can allocate computing tasks to multiple computing nodes, so as to solve the problem of parallel processing. Task execution means that computing nodes perform the tasks assigned to them and return the results to the master node.

1.2.2. Knowledge Integration

Through the data mapping technology, the mapping relationship between the terms in the ontology and the vocabularies extracted from different data sources for knowledge is established, and then the data from different data sources are integrated together [8]. There are also some terms describing the same kind of data between different ontologies. So for these ontologies, ontology fusion technology is needed to fuse different ontologies. Finally, the integrated knowledge base needs a storage and management solution.

1.2.3. Knowledge Computing

Knowledge computing refers to the process of deriving more implicit knowledge from the information provided by the graph. For example, through ontology, Knowledge computing can obtain the implicit knowledge [9]. At the same time, knowledge computing uses different algorithms of social computing to calculate on the knowledge network. Calculating the knowledge graph through knowledge computing can generate a large number of intelligent applications. For example, it can provide accurate user portraits, provide domain knowledge to expert systems, and provide decision data. It also can provide a more intelligent retrieval method so that users can search through natural language.

1.2.4. Entity Alignment

The traditional method only considers the attributes of each entity, but does not consider the relationship between entities. It aligns entities by evaluating various similarities. Entity alignment based on knowledge representation can use the structural features and semantic features of the upper and lower levels. The entity alignment of knowledge graph based on Neo4j graph database is the most commonly used method at present. It uses node attributes or labels to represent entity names [10]. First, it adds entities from different data sources to the graph database and sets the corresponding attributes. Then, by executing the Cypher query statement, it finds similar entity names and aligns them. The same entity will have different references in the text.

2. Constructing Mathematical Models

Constructing a mathematical model is the core work of this paper, including extracting word vectors, calculating user interest preference features, generating news feature vectors, convolutional neural network recommendation models, and integrating attention propagation layers. In this paper, the recurrent neural network model integrated with the attention mechanism is defined as ANCNN.

2.1. Extracting Word Vectors

In order to convert the words in news headlines and abstracts into low-dimensional dense vectors with rich semantics, this paper uses the unsupervised learning method in the Word2Vec model to obtain the corresponding word vectors. It utilizes a large corpus of sentences as a training set, simulating the human brain's thinking through the form of a neural network, encoding words with one-hot vectors as input, and training the model to achieve a vectorized representation of text data [11]. After the words in the news text are mapped to the semantic vector space, the mathematical operation can be carried out between the vectors. The distance reflects the relevance of words. In order to fully extract the features of news text, this paper initially converts the words in news headlines and news abstracts into semantic vectors extracted after corpus data training.Let m denote the length of the news headline, let n denote the length of the news summary, let $S_t=[w_1,w_2,\cdots,w_m]$ denote the word sequence of a news headline, and let $S_a=[w_1,w_2,\cdots,w_n]$ denote the word sequence of a news summary. The word vector extraction process of news text is shown in Figure 2.

DOI: 10.55214/25768484.v9i5.7419

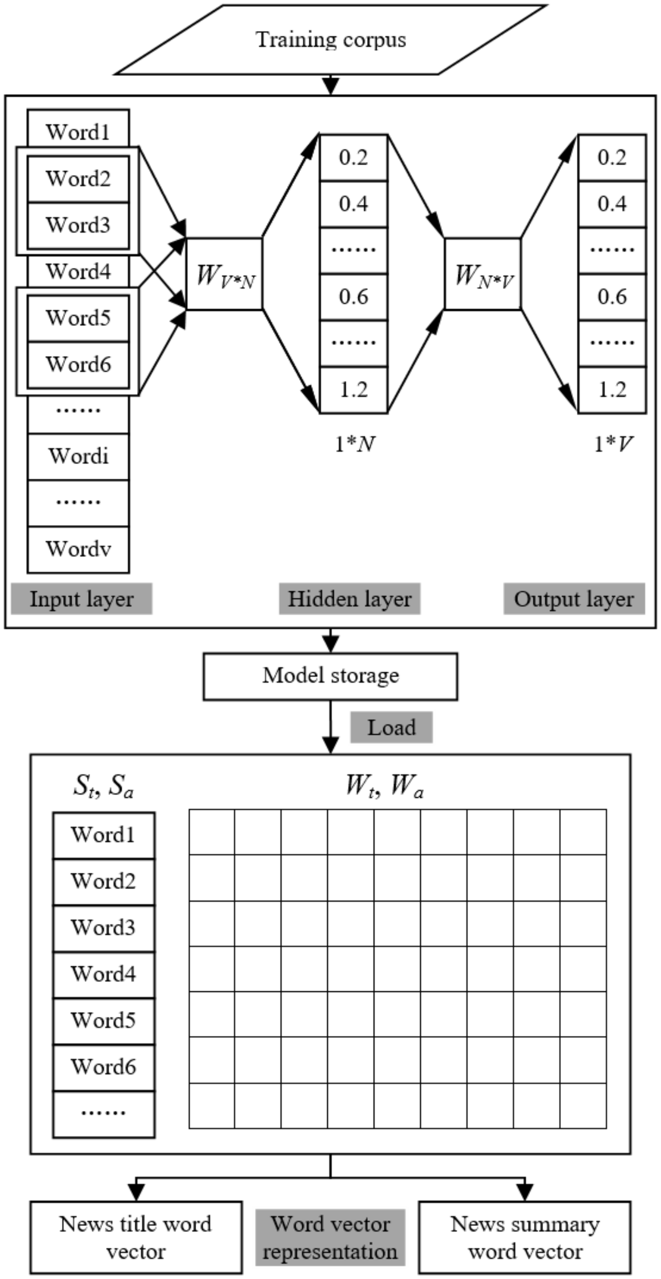

Figure 2.
Extraction process of news text word vector.

Text training. For sentences in the corpus, this paper selects a sliding window with a length of $2C+1$, that is, the context window of the corpus words is C, and generates training samples. According to the hypothesis of the neural network model used to generate word vectors, the selection of each word is determined by adjacent words. The coding vector of context words is used as the input of the model. With the help of gradient descent method, the loss function is recursively trained by sample data to determine and store the optimal hidden layer weight matrix.

Semantic vector generation. The optimal weight matrix is loaded to generate the feature vector corresponding to each word, and the word vector matrix of the news text is obtained. The formula is as follows :

$$W_t = \left[w_1^t, w_2^t, \cdots, w_i^t \cdots, w_m^t \right] \qquad (1)$$

$$W_a = \left[w_1^a, w_2^a, \cdots, w_i^a \cdots, w_n^a \right] \qquad (2)$$

In the above formula, W_t represents the news title word vector matrix, W_a represents the news summary word vector matrix, and w_i^t and w_i^a represent the word vectors generated by processing in the text.

2.1.1. Calculating User Interest Preference Characteristics

There are a large number of click behaviors in the user history browsing sequence. There is a certain correlation between most of the user's historical behaviors, and the user's current behavior may be affected by other behaviors. Therefore, the use of multi-layer attention networks can effectively solve this problem. Specifically, this paper uses a multi-layer attention network to calculate the correlation between behaviors, and determines the degree of influence of other behaviors according to the size of the weights.

During browsing, users may be interested in a variety of news types. This has a certain impact on predicting whether users click on a specific type of news. Users' browsing behavior has different degrees of influence on each news headline. Therefore, for the user representation of the current candidate news, this paper uses a multi-layer attention network to rate the user's click behavior. The rating will be used as an important indicator for screening user behavior. In order to represent the user's interest preference characteristics, the user's current interest is expressed as:

$$e\left(t_k^i\right) = \tanh\left(W_w e\left(t_m^i\right) + b_w\right) \qquad (3)$$

The user 's other behaviors are expressed as:

$$e\left(t_j\right) = \tanh\left(W_t e\left(t_j\right) + b_t\right) \qquad (4)$$

The weight size processed by the attention network is expressed as:

$$\begin{aligned} s_j &= soft\max\left(H\left(e\left(t_k^i\right), e\left(t_j\right)\right)\right) \\ &= \frac{\exp\left(H\left(e\left(t_k^i\right), e\left(t_j\right)\right)\right)}{\sum_{k=1}^{N_i} \exp\left(H\left(e\left(t_k^i\right), e\left(t_j\right)\right)\right)} \end{aligned} \qquad (5)$$

The user's current interest and other behaviors are processed by the attention mechanism to obtain the weight between the two behaviors. Then this paper judges the correlation between the two behaviors by the weight size. According to the weight size, the user's dynamic interest characteristics are calculated. The calculation formula is:

$$e(i) = \sum_{k=1}^{N_i} s_j e\left(t_k^i\right) \qquad (6)$$

2.1.2. Generate News Feature Vectors for Title

User interest modeling consists of three steps: The first step is to obtain the representation of each news text feature based on the user 's previous reading news. The second step is to obtain the representation of the entity knowledge vector and the representation of the context vector of the news text through knowledge graph representation learning. These representations are used to compute the feature representations of the news. The third step, according to the feature representation of the user's historical reading news, this paper obtains the user's interest feature representation through the gated recurrent unit network. The news feature vector generation process is shown in Figure 3.

For the same entity, there may be multiple entities that have the same relationship with it. According to the vector representation of the triplet relationship, adjacent entities will have similar vector representations under the same relationship. Therefore, this paper uses the head entity vector plus the tail entity vector as a tail entity set that has a similar relationship with the head entity. It can also be said that the tail entity set is the context information of the head entity. In the graph structure, let *e* denote the entity, *r* denote the relationship, *R* denote the set of all relationships in the KG, and the context set of the entity is represented as:

$$context\left(e\right) = \left\{e + r_i \middle| r_i \in R\right\} \qquad (7)$$

The more the number of entities in the set of adjacent entities, the more difficult it is to calculate. Therefore, in order to facilitate the representation of the context information of the entity, this paper needs to simplify the expression of the context information of the entity. By calculating the average value of the context entity, this paper takes it as the entity context vector. The calculation formula is:

$$\bar{e} = \frac{1}{\left|context\left(e\right)\right|} = \sum_{e_i \in context(e)} e_i \qquad (8)$$

For a news text, there are three aspects of information: the keyword sequence of the news itself, the vector representation of the entity, and the context representation of the entity. In this paper, three *embedding* matrices are aligned and superimposed to obtain multi-channel word representations. These representations are sent to the convolutional neural network for training to extract spatial structure information. Finally, the representation vector of news is obtained.

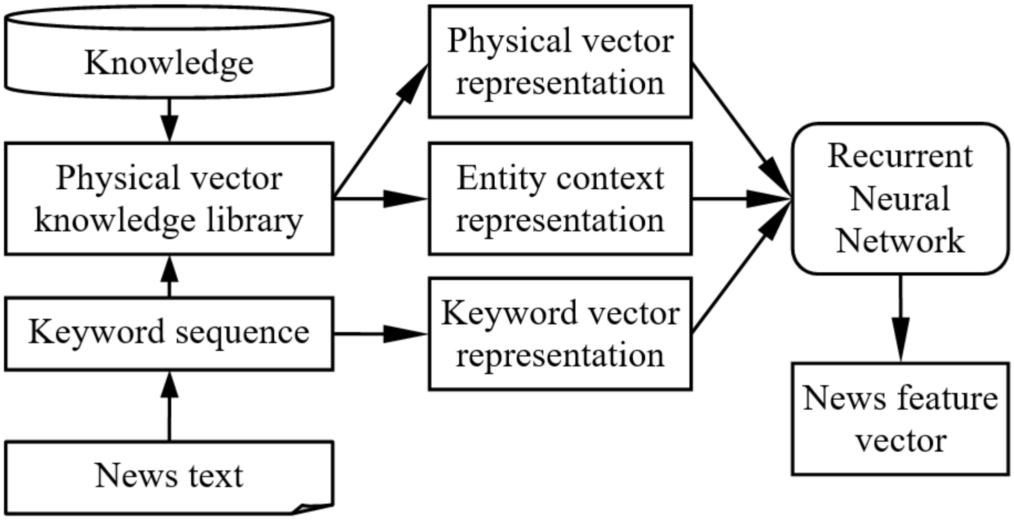
Knowledge
Physical vector knowledge library
Keyword sequence
News text
Physical vector representation
Entity context representation
Keyword vector representation
Recurrent Neural Network
News feature vector

Figure 3.
News feature vector generation.

2.1.3. Convolutional Neural Network Recommendation Model

The calculation formula is :

$$c_i = f\left(w \square X_{i,i+h-1} + b\right) \qquad (9)$$

In the above formula, c_i represents the i-th eigenvalue in the characteristic graph.Thus, the obtained feature vector is:

$$C=\left[c_1,c_2,\cdots,c_{n-h+1}\right] \qquad (10)$$

In this paper, we use the maximum pooling method to sample features, with a pooling window of 2×1 and a horizontal movement step size of 2. After the pooling operation is completed, the dimension of the feature vector becomes half of the original, expressed as:

$$C_p=\left[c_1,c_2,\cdots,c_{(n-h+1)/2}\right] \qquad (11)$$

After the pooling layer, a fully connected layer is connected, and the extracted feature vectors are passed into the fully connected layer for fitting. Since the convolutional neural network generally has *m* convolution kernels for feature extraction, the corresponding m feature vectors will be generated. After the *m* feature vectors are recombined, the input vector of the fully connected layer is obtained, which is expressed as:

$$V=\left[c_{1,1},\cdots,c_{j,i},\cdots,c_{m,(n-h+1)/2}\right] \qquad (12)$$

In the above formula, $c_{j,i}$ denotes the *i*-th eigenvalue of the *j*-th eigenvector. In order to prevent over fitting, a dropout processing can be added on the full connection layer. Dropout is a random deactivation technique. During the training process, Dropout randomly deletes some neurons, so that the model has a different structure in each iteration. This helps to improve the generalization ability of the model and reduce the dependence on training data. Finally, the output of the fully connected layer is passed into the softmax function to calculate the probability of each news being recommended and output.

After the completion of the model construction, training is needed. The loss when training the model is measured by cross entropy, and the loss function is defined as:

$$loss=-\sum_{i=1}^{n}y_i'\log\left(y_i\right) \qquad (13)$$

In the above formula, *n* represents the number of news.□ y_i' indicates whether the *i*-th news is really liked by the current user. It is 1 for likes and 0 for dislikes. y_i is the probability that the model predicts that the *i*-th news is liked by the current user.

2.1.4. Incorporating the Attention Propagation Layer

Graph neural network and attention mechanism recursively perform propagation on the graph and embed entities, which can achieve good recommendation results [12, 13]. The attention propagation layer is implemented through message propagation and message aggregation [14, 15]. Message propagation first calculates the linear combination of entities *h* in the network (*ego* network):

$$e_{Nh}=\sum_{(h,r,t)\in Nh}\pi\left(h,r,t\right)e_t \qquad (14)$$

The $\pi\left(h,r,t\right)$ attention score represents the amount of information transmitted from *t* to *h* under the condition of relationship *r*. The calculation formula is:

$$\pi\left(h,r,t\right)=\left(W_re_t\right)^T\tanh\left(W_re_h+e_r\right) \qquad (15)$$

The attention score depends on the distance between e_h and e_r in the relation *r* space.

Message aggregation can use *GCN* aggregation or *GS* aggregation. Let *W* be the training parameter, *GCN* aggregation and *GS* aggregation are expressed as :

DOI: 10.55214/25768484.v9i5.7419

$$f_{GCN} = \text{LeakyReLU}\left(W\left(e_h + e_{Nh}\right)\right) \qquad (16)$$

$$f_{GS} = \text{LeakyReLU}\left(W\left(e_h \| e_{Nh}\right)\right) \qquad (17)$$

3. Simulation Experiment

The ANCNN recommendation will be compared with the commonly used benchmark. The experimental data are from the Adressa dataset.

3.1. Experimental Data

In the field of news recommendation, Adressa is the best data set at present. The Adressa dataset was jointly released by Norwegian news publishers such as Adresseavisen. It is part of RecTech's recommended technology project [16]. There are cases where multiple non-subscribers are the same user in the dataset, and there are also cases where the data is sparse. The Adressa dataset consists of two parts, a list of users' readings and a list of articles. The Adressa dataset contains implicit feedback content such as clicks and reading time. Although it lacks display feedback, it is sufficient to complete the training task of the user recommendation system.

The Adressa dataset is divided into two subsets: Adressa-1week and Adressa-10wveeks.The Adressa-1week dataset contains data on users ' browsing news within a week. The Adressa-10weeks dataset contains data on users browsing news within 10 weeks. For each event, this article uses seven attributes, namely, session start, session end and profile. These seven attributes are used to generate data sets and sort the news according to the timestamp. The Adressa dataset is shown in Table 1.

Table 1.
Overall information about the news data set.

No	Category	Adressa-1week	Adressa-10week
1	Number of users	537627	590673
2	Number of news	14732	49994
3	Number of events	2527571	23168411
4	Number of entities	116603	2792114
5	The number of entities in the context of the entity	43.1	47.2

3.2. Evaluating Indicator

The generalization ability of the evaluation model requires both a scientific and rigorous experimental process and appropriate evaluation indicators. There are many common evaluation indexes. Therefore, different evaluation indicators should be selected for specific data distribution, otherwise wrong conclusions may be drawn. This paper uses F_1 and AUC two evaluation indicators.

Recall indicates that it recommends news that users are interested in, but it also recommends news that users are not interested in.

$$Recall = \frac{TP}{TP + FN} \qquad (18)$$

Precision indicates that the news that the user is interested in is recommended, but the news that the user is interested in is not all recommended.

$$Precision = \frac{TP}{TP + FP} \qquad (19)$$

F_1 combines *Precision* and *Recall.* As both have their respective flaws and are in a state of mutual suppression and growth. Only when both are high, F_1 will be high.

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (20)$$

AUC value is a probability value. The calculation formula is to calculate the rectangular area under the curve:

$$AUC = \sum_{i=2}^{m} \frac{(x_i - x_{i-1}) \times (y_i - y_{i-1})}{2} \qquad (21)$$

The meaning expressed by the *AUC* indicator, simply explained, is that a pair of samples is randomly selected, consisting of one positive sample and one negative sample. Then, the classifier obtained through training is used to make predictions for these two samples. The probability that the predicted probability of the positive sample is greater than that of the negative sample is what the *AUC* represents. The calculation formula of *AUC* is:

$$AUC = \frac{\sum_{i \in positiveCalss} rank_i - \frac{M(1+M)}{2}}{M \times N} \qquad (22)$$

3.3. Environment and Parameter Configuration

In order to carry out simulation experiments, it is necessary to complete the environment and parameter. The environment includes hardware environment and software environment. Parameter is to configure the parameters of the recurrent neural network. Some parameters may be adjusted as needed during the experiment. The initial environment and parameter configuration are shown in Table 2.

Table 2.
Environment and parameter configuration.

No	Name	Configuration
1	Operating system	Ubuntu16.04
2	CPU	Core i7-980X
3	GPU	NVIDIA Tesla V100
4	Memory	DDR5 6800 32GB
5	Hard drive	1TB SSD
6	Development tool	Jet Brains PyCharm
7	Deep learning framework	TensorFlow 2.6.0
8	Compile environment	Python 3.8
9	optimizer	Adam algorithm
10	epochs	30
11	Loss function	Cross Entropy Loss
12	weight_decay	0.001
13	scheduled_sampling_prob	0.5

3.4. Selection of Baseline Model

In order to verify the advantages of the ANCNN model, this paper selects four benchmark models such as "DeepFM, DKN, NAML, and NPA". The four benchmark models are briefly described as follows:

(1) DeepFM [17] is to change the Wide & Deep Wide part to FM, simply expressed as FM + DNN. The Wide part and the Deep part share the input Embedding vector. The parameters are updated by back propagation through high-order and low-order interactive features, so that the model performs better. The Deep part is a full link layer, which is used to learn some high-level feature interactions.

(2) DKN, DKN has made substantial gains on the most advanced deep recommendation model. DKN uses entity information of knowledge graph to enhance news features, and learns entity embedding through knowledge subgraph and distance-based translation model. The DKN model uses the attention mechanism to calculate user interest and improves the accuracy of recommendation.

(3) NAML, NAML includes two core parts. NAML takes the title, text and category as different perspectives of news to learn a unified news representation. Due to the different amount of information represented by different words and views in the news. In the user encoder, the user representation is learned from the representation of the user clicking on the news.

(4) NPA, through the personalized attention mechanism, improves the user's reading experience. The core of NPA lies in two models, which are news representation and user representation. Among them, the news representation model extracts key information that can attract users' attention through deep mining of news content. The user representation learns the user's personalized features from the news that the user has clicked. This provides rich context information for the model and makes the recommendation more accurate. At the same time, NPA adopts two levels of attention mechanism. Word-level attention helps the model focus on keywords that different users are interested in, and news-level attention focuses on the importance of news that users have clicked historically.

3.5. Experimental Results and Analysis

The ANCNN model proposed in this paper is compared with the four benchmark models of "DeepFM, DKN, NAML, NPA", and shown in Table 3.

Table 3.
Performance comparison.

Model	Adressa-1week		Adressa-10week	
	AUC	***F_1***	***AUC***	***F_1***
DeepFM	73.89	72.18	72.46	71.35
DKN	72.47	69.59	71.15	67.71
NAML	78.64	76.86	77.47	75.28
NPA	81.32	80.12	79.44	78.03
The average of the benchmark model	76.58	74.69	75.13	73.09
ANCNN (This paper)	82.66	81.37	80.92	80.54
Improvement over the benchmark model	6.08	6.68	5.79	7.45

According to the data of table 3, the comprehensive analysis is as follows:

(1) For the four benchmark models "DeepFM, DKN, NAML, NPA", the "Adressa-1week" dataset and the "Adressa-10week" dataset are consistent. The order from good to bad is "NPA, NAML, DeepFM, DKN", and the NPA model is optimal. However, the ANCNN model is superior to the NPA model.

(2) The optimal model of the four benchmark models is the NPA model. Compared with the NPA, the ANCNN proposed in this paper achieves an absolute increase of 1.35 in the AUC metric and a relative increase of 1.65% for the "Adressa-1week" dataset. The absolute increase of the F1 metric is 1.25 and the relative increase is 1.56%.For the "Adressa-10week" dataset, the absolute value of AUC index increased by 1.48%, and the relative value increased by 1.86%. The absolute value of F1 index increased by 2.51, and the relative value increased by 3.22%.

(3) The worst model of the four benchmark models is DKN model. Compared with the DKN, the absolute value of the AUC index is increased by 10.19 and the relative value is increased by 14.06% for the "Adressa-1week" dataset. The absolute value of F1 index increased by 11.78, and the relative value increased by 16.93%. For the "Adressa-10week" dataset, the absolute value of AUC index increased by 9.77, and the relative value increased by 13.73%. The absolute value of F1 index increased by 12.83, and the relative value increased by 18.95%.

(4) The ANCNN model in this paper is the most convincing compared with the average of the four benchmark models. For the "Adressa-1week" dataset, the absolute value of AUC index increased by 6.08, and the relative value increased by 7.94%. The absolute value of F1 index increased by 6.68, and the relative value increased by 8.95%. For the "Adressa-10week" dataset, the absolute value of AUC index increased by 5.79, and the relative value increased by 7.71%. The absolute value of F1 index increased by 7.45, and the relative value increased by 10.19 %.

The four benchmark models of "DeepFM, DKN, NAML, NPA" are all based on CNN to extract news features from the original word vectors. The ANCNN model proposed in this paper is further improved on the basis of CNN, which is mainly reflected in the use of graph attention network to weight each neighborhood entity. It extracts entities based on the size of their weights, to some extent enhancing the information in entity vectors. Moreover, HNCNN adds a multi-layer attention network on the basis of CNN model, which can more effectively extract the correlation between user behaviors. When extracting neighborhood information, the ANCNN model is more targeted for relevance search. And the ANCNN model also improves the attention network when performing user interest extraction, making the user representation more accurate. Therefore, the ANCNN model is better than the four benchmark models in news recommendation.

4. Conclusion

Facing the news characteristics and user needs in the era of media convergence, news recommendation can help users push news of interest from massive news, which improves the efficiency of news selection and reading experience. Knowledge graph can deeply mine and analyze entities and relationships. Compared with traditional recommendation methods, the recommendation based on KG can capture rich semantic and structural information, analyze user's personalized preferences more accurately. It also has the advantages of interpretability and expansibility. The research results help users find news information that matches their needs, effectively solve the user loss caused by excessive information in the era of media convergence, and reduce the complexity and training time of the model. Future research will continue to explore the capture of knowledge graph information and the accurate expression of embedding vectors, and mine the evolution law of knowledge graph data. Future research attempts to use more advanced information aggregation and message propagation methods to optimize the model, further improving the recommendation quality and the robustness of the recommendation algorithm. At present, the impact of position deviation can not be eliminated well in news recommendation. If adversarial learning is added to the news recommendation model, it can effectively eliminate the impact of location bias and obtain accurate user feature representation. In the future development, this direction is very promising.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

References

[1] L. M. Kristensen and P. Bro, "News values in a digital age-Intra-media, inter-media, and extra-media platforms," *Journalism*, vol. 25, no. 4, pp. 819-836, 2024. https://doi.org/10.1177/14648849231162266

[2] K. Tilman, P. Fabian, and S. Wolfgang, "Public knowledge of alternative media in times of algorithmically personalized news," *New Media & Society*, vol. 25, no. 7, pp. 1648–1667, 2023. https://doi.org/10.1177/14614448221144750

[3] P. Ruenin, M. Choetkiertikul, A. Supratak, and S. Tuarob, "TeReKG: A temporal collaborative knowledge graph framework for software team recommendation," *Knowledge-Based Systems*, vol. 289, p. 111492, 2024. https://doi.org/10.1016/j.knosys.2024.111492

[4] M. Ismail, S. Alrabaee, and R. K. K. Choo, "A comprehensive evaluation of machine learning algorithms for web application attack detection with knowledge graph integration," *Mobile Networks and Applications*, *Advance online Publication*, pp. 1–30, 2024. https://doi.org/10.1007/s11036-024-02250-5

[5] H. N. Cuo, Z. Q. Sun, and W. Hu, "A pre-trained universal knowledge graph reasoning model based on rule prompts," *Journal of Computer Research and Development*, vol. 61, no. 8, pp. 2030–2044, 2024.

[6] K. Gurinder, L. Fei, and Y. C. Phoebe, "A deep learning knowledge graph neural network for recommender systems," *Machine Learning with Applications*, vol. 14, no. 1, pp. 1–19, 2023. https://doi.org/10.1016/j.mlwa.2023.100401

[7] H. Kim and J. Choi, "Recommendations for responding to system security incidents using knowledge graph embedding," *Electronics*, vol. 13, no. 1, pp. 1-19, 2023. https://doi.org/10.3390/electronics13010001

[8] Z. Aizan, S. K. Sahoo, D. Varshney, A. Das, and A. Ekbal, "KIMedQA: Towards building knowledge-enhanced medical QA models," *Journal of Intelligent Information Systems*, vol. 62, no. 3, pp. 833-858, 2024. https://doi.org/10.1007/s10844-024-00784-y

[9] M. A. Khan, S. U. R. Khan, S. Z. Q. Haider, S. A. Khan, and O. Bilal, "Evolving knowledge representation learning with the dynamic asymmetric embedding model," *Evolving Systems*, vol. 15, no. 6, pp. 2323-2338, 2024. https://doi.org/10.1007/s12530-024-09535-4

[10] J. Monteiro, F. Sá, and J. Bernardino, "Experimental evaluation of graph databases: Janusgraph, nebula graph, neo4j, and tigergraph," *Applied Sciences*, vol. 13, no. 9, p. 5770, 2023. https://doi.org/10.3390/app13094447

[11] S. Sukri, N. A. Samsudin, E. Fadzrin, S. K. Ahmad Khalid, and L. Trisnawati, "Word2vec-based latent semantic indexing (Word2Vec-LSI) for contextual analysis in job-matching application," *International Journal of Advanced Computer Science & Applications*, vol. 15, no. 3, pp. 699–707, 2024.

[12] F. Torrisi, C. Corradino, S. Cariello, and C. Del Negro, "Enhancing detection of volcanic ash clouds from space with convolutional neural networks," *Journal of Volcanology and Geothermal Research*, vol. 448, p. 108046, 2024. https://doi.org/10.1016/j.jvolgeores.2023.108046

[13] A. Kurniawan, E. Erlangga, T. Tanjung, F. Ariani, Y. Aprilinda, and R. Y. Endra, "Review of deep learning using convolutional neural network model," *Engineering Headway*, vol. 3, pp. 49-55, 2024.

[14] G. Dagmar, B. Russa, and S. Harald, "MADLINK: Attentive multihop and entity descriptions for link prediction in knowledge graphs," *Semantic Web*, vol. 15, no. 1, pp. 83-106, 2024. https://doi.org/10.3233/SW-230503

[15] J. Y. Xing, X. Xing, and Z. C. Jia, "Item recommendation algorithm integrating knowledge graph and attention mechanism," *Journal of Computer Engineering & Applications*, vol. 60, no. 10, pp. 173–179, 2024.

[16] M. A. Rauf, M. M. Y. Khalil, M. A. N. U. Ghani, W. Wang, Q. Wang, and J. Hassan, "ZS-CEBE: Leveraging zero-shot cross and bi-encoder architecture for cold-start news recommendation," *Signal, Image and Video Processing*, vol. 18, no. 8, pp. 6455-6467, 2024. https://doi.org/10.1007/s11760-024-03648-7

[17] K. F. Hu and W. D. Wang, "DeepFM recommendation model based on encoder enhancement," *Computer & Digital Engineering*, vol. 53, no. 1, pp. 158–163, 2025.

DOI: 10.55214/25768484.v9i5.7419