


Article

# News Recommendation Based on User Topic and Entity Preferences in Historical Behavior

Haojie Zhang<sup>1</sup> and Zhidong Shen<sup>1,2,\*</sup> 

<sup>1</sup> Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430079, China

<sup>2</sup> Engineering Research Center of Cyberspace, Yunnan University, Kunming 650091, China

\* Correspondence: shenzd@whu.edu.cn

**Abstract:** A news-recommendation system is designed to deal with massive amounts of news and provide personalized recommendations for users. Accurately modeling of news and users is the key to news recommendation. Researchers usually use auxiliary information such as social networks or item attributes to learn about news and user representation. However, existing recommendation systems neglect to explore the rich topics in the news. This paper considered the knowledge graph as the source of side information. Meanwhile, we used user topic preferences to improve recommendation performance. We proposed a new framework called NRTEH that was based on topic and entity preferences in user historical behavior. The core of our approach was the news encoder and the user encoder. Two encoders in NRTEH handled news titles from two perspectives to obtain news and user representation embedding: (1) extracting explicit and latent topic features from news and mining user preferences for them; and (2) extracting entities and propagating users' potential preferences in the knowledge graph. Experiments on a real-world dataset validated the effectiveness and efficiency of our approach.

**Keywords:** news recommendation; knowledge graph; topic preference; historical behavior



**Citation:** Zhang, H.; Shen, Z. News Recommendation Based on User Topic and Entity Preferences in Historical Behavior. *Information* **2023**, *14*, 60. <https://doi.org/10.3390/info14020060>

Academic Editors: Pierpaolo Basile and Annalina Caputo

Received: 10 November 2022

Revised: 3 January 2023

Accepted: 3 January 2023

Published: 18 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

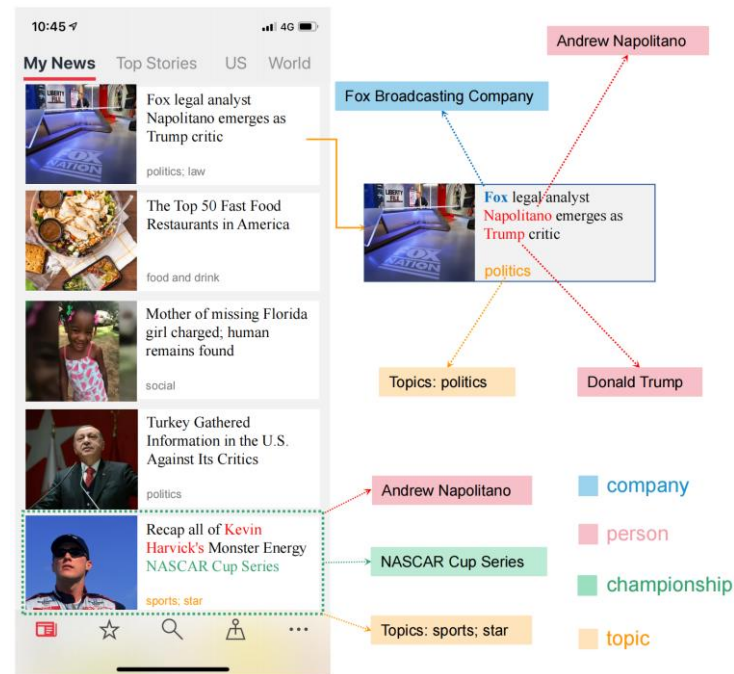
## 1. Introduction

With the development of the Internet, people's news-reading habits have gradually shifted from traditional media such as newspapers and television to the Internet. Online news websites collect news contents from a variety of sources and provide them to users, which attracts a large number of users. However, due to the large amount of news generated every day, it is impossible for users to read all the articles. Although news articles cater to different personal interests, it is often difficult for users to manually select interesting articles that correspond to their own interests. Therefore, it is critical to help users target their reading interests and make personalized recommendations [1–6].

In order to improve the accuracy of recommendation systems, recent research focused on learning the representation of news more comprehensively. For example, Wang et al. [3,4] utilized knowledge graphs (KGs) to obtain explain ability and informative connections among the items recommended. A deep knowledge-aware network (DKN) [3] embeds each news title from three perspectives: word, entity, and entity context, then designs a CNN model to aggregate these features together. RippleNet [4] obtains the potential interest of users by automatically and iteratively spreading their preferences in the knowledge graph. However, DKN and RippleNet not only ignore rich semantic topics in the news titles, but also do not consider the relevance between topics and users' preferences for those topics to learn more precise news representations. Topic information is also a vital factor to attract users' interests and can improve the efficiency of the recommendation model if well employed.

As shown in Figure 1, news titles may contain a variety of entities such as politicians, celebrities, companies, or institutions; these entities are not independent from one another

but can be linked to other entities through various relationships and are organized as a graph. Meanwhile, news titles may also contain multiple topics such as politics, entertainment, sports, etc., which often play an important role in the title as along with the entities.



**Figure 1.** Illustration of news title with a variety of entities and topics.

LSTUR [1] uses explicitly given topic information to learn the representation of news titles. Although explicit topic labels can accurately represent the information in the news, when a news title contains two or more different topics, simple topic information may not be detailed enough to give a more comprehensive representation of the news topic. Therefore, the latent topic information is needed to model the news titles in more detail.

For example, the news title “Trump rule may mean 1 million kids lose automatic free lunch” appears as a health topic. However, the content of the news is more relevant to politics. Such misinterpretation in news modeling can lead to serious errors in learning users’ topic preferences. Therefore, only considering the explicit topic information and ignoring latent topic information of news will reduce the accuracy of news-recommendation systems.

To address the limitations of existing methods and inspired by the wide success of leveraging knowledge graphs, in this paper, we proposed a news-recommendation approach based on topic and entity preference in historical behavior. NRTEH was designed for click-through rate (CTR) prediction, which takes a piece of candidate news and a user’s click history as input and then outputs the probability of the user clicking the news titles. In this paper, we only used news titles as input because news titles are a key factor in attracting users to read them.

The core of our approach was a news encoder and a user encoder. In the news encoder, we jointly trained news title and word vectors to obtain the topic information of the news and extract entities from a knowledge graph. In the user encoder, we used a combination of a long short-term memory network and a self-attention mechanism to mine the users’ topic preferences together with a graph attention algorithm to mine the users’ potential preferences for the entities in knowledge graph based on the users’ historical behavior.

Extensive experiments on a real-world dataset proved the validity of our news recommendation method. The results showed that NRTEH achieved significant improvements over state-of-the-art deep-learning-based methods in news recommendation. Specifically, NRTEH notably outperformed the baselines by 2.6% to 11.4% based on the AUC and 3.3% to 12.4% based on the ACC. The results also demonstrated that the usage of contextual

knowledge and latent topic classification could result in additional improvements in the NRTEH framework.

## 2. Related Work

News recommendation has been widely studied previously. Unlike other recommendation domains, news recommendation is much more complicated in text mining. In general, because news items are constantly replaced, the traditional CF approaches [2,7] commonly suffer from several fundamental issues such as data sparsity and cold-start problems. To address these issues, the researchers used content-based techniques to provide complementary information to CF [6,8–11]. For example, Jeong et al. [6] proposed an explicit localized semantic analysis (ELSA) method for news recommendation in which every location had its own geographical topics that could be captured from the geo-tagged documents related to the location. Okura et al. [8] proposed to learn news embeddings based on the news similarities while considering topic information and using recurrent neural networks (RNNs) to learn user representations from their click histories. Wang et al. [9] proposed a novel module named the Feature Refinement Network (FRNet), which learned context-aware feature representations at the bit level for each feature in different contexts. However, while these methods provided additional information to improve the accuracy of recommendation systems, they did not consider users' long-term or short-term preferences in news reading.

In recent years, recommendation systems have been taking greater advantage of deep learning to achieve better performance in various recommendation scenarios. Generally speaking, deep neural networks can be used in deep recommendation systems in two forms: modeling the interaction among users and items or processing the raw features of users or items. These methods include DSSM [12], DeepFM [13], DMF [14], DeepWide [15], and DeepGroup [16] models. In addition, Multiview Deep Learning [17], SHINE [18], and WG4Rec [19] are also popular deep-learning-based recommendation systems.

In addition to recommendation systems based on deep learning, some researchers have added KGs as side information in order to more accurately obtain users' preferences for target items. KGs such as YAGO [20], Freebase [21], and Google Knowledge Graph [22] contain entities such as places, people, and organizations from real-world objects and concepts. In order to introduce a knowledge graph into a recommender system, knowledge graph representation has also been studied extensively recently. The purpose of knowledge graph representation is to determine a low-dimensional vector for each entity and relation in the knowledge graph while preserving the structure of the original graph. For example, KGAT [23] utilizes a self-attention network for information propagation and a multi-head attention mechanism to increase model capacity, while a graph convolutional network utilizes localized graph convolutions in classification tasks. In recent years, translation-based knowledge-graph-embedding methods such as TransE [24], TransH [25], TransD [26], TransR [27], and LightKG [28] have attracted wide attention due to their simple models and superior performances. They utilize distance-based scoring functions to determine representations of entities and relations.

Because a knowledge graph can expand item information to a large extent, it can also be used in many applications such as question answering [29], text classification [30], and even in regional similarity searches [31]. Different from these application scenarios, a news-recommendation model needs to deal with semantic connections between different types of entities in KGs, so news text mining is the key to utilizing entities contained in news. For example, a DKN [3] sends knowledge representation of entities contained in news titles to a CNN model as a channel to learn users' entity preferences. RippleNet [4] uses an attention mechanism to automatically propagate the clicked entities in the knowledge graph to capture the higher-order preferences of users. KGCN [32] utilizes a knowledge graph convolutional network to extend the non-spectral graph convolutional network [33–35] method to a knowledge graph through selectively aggregating the neighborhood node features. KAeDCN [36] combines the dynamic convolutional network with attention mechanisms

to capture changing dependencies in the sequence and enhance the representations of items with KG information through an information-fusion module to capture fine-grained user preferences.

Therefore, in our approach, we used an external knowledge graph and deep neural networks to model user and news representations to consider both issues using CF approaches and content-based approaches. The major difference between these methods and ours is that we used a new method to automatically extract latent topics in the news and then used a memory neural network and a self-attention mechanism to learn users' topic preferences in historical behavior. We considered not only user preferences for news topics, but also user interest in the entities included in the news. We used a graph attention mechanism to learn the entities that the users preferred. In short, by contemplating different levels of news, our goal was to model news and users as accurately as possible.

### 3. Problem Formulation

We formulated the news-recommendation problem in this paper as follows. The inputs of the system were the candidate news title and clicked news titles, while the output was the probability of the user clicking the candidate news. For a given user  $u_i$  in the online news platform, we denoted their click history as  $\{t_1^i, t_2^i, \dots, t_{N_i}^i\}$ , where  $t_j^i (j = 1, \dots, N_i)$  is the title of the  $j$ -th news clicked by user  $u_i$ , and  $N_i$  is the total number of user  $u_i$ 's clicked news titles. We also implemented knowledge graph  $G$ , which contained entity–relation–entity triples  $(h, r, t)$ . Note that  $h, r$ , and  $t$  are the head, relation, and tail, respectively, of a triple in a KG. The user–news interaction matrix  $Y = \{y_{ut} | u \in U, t \in T\}$  was defined according to user's clicked history as:

$$y_{ut} \begin{cases} 0, & \text{if user } u \text{ did not click news } t \\ 1, & \text{if user } u \text{ clicked news } t \end{cases} \quad (1)$$

Each news title  $t$  was composed of a sequence of words; i.e.,  $t = [w_1, w_2, \dots]$ , where each word  $w$  may be associated with an entity  $e$  in the knowledge graph. For example, in the title "Fox legal analyst Napolitano emerges as Trump critic", "Napolitano" is linked with the entity "Andrew Napolitano", while "Trump" is linked with the entity "Donald Trump". In addition, we also extracted explicit and latent topic-level connections among the news titles. Given the users' click history as well as the topics in news titles and entities in the knowledge graph, we aimed to predict whether user  $u_i$  would click a candidate news  $t_j$  that they had not seen before.

### 4. Preliminaries

In this section, we present several concepts related to this work, including knowledge-graph embedding, the triple set, doc2vec, and self-attention.

#### 4.1. Knowledge-Graph Embedding

A typical knowledge graph consists of millions of entity–relation–entity triples  $(h, r, t)$  in which  $h, r$ , and  $t$  represent the head, the relation, and the tail of a triple, respectively. Given all the triples in a knowledge graph, the goal of knowledge-graph embedding is to determine a low-dimensional representation vector for each entity and relation that preserves the structural information of the original knowledge graph. We studied and analyzed the advantages and disadvantages of existing methods such as TransE [24], TransH [25], TransD [26], TransR [27], and LightKG [28] and finally chose TransE as the knowledge representation learning algorithm of our model.

TransE wants  $h + r \approx t$  when  $(h, r, t)$  holds, where  $h, r$ , and  $t$  are the corresponding representation vectors of  $h, r$ , and  $t$ . Therefore, TransE assumes that the score function

$$f_r(h, t) = \left\| \left| h + t - r \right| \right\|_2^2 \quad (2)$$

is low if  $(h, r, t)$  holds, and high otherwise.

To encourage the discrimination between correct triples and incorrect triples, the following margin-based ranking loss was used for training:

$$L = \sum_{(h,r,t) \in \Delta} \sum_{(h',r',t') \in \Delta'} \max(0, f_r(h, t) + \gamma - f_r(h', t')) \quad (3)$$

where  $\gamma$  is the margin and  $\Delta$  and  $\Delta'$  are the respective sets of correct triples and incorrect triples.

#### 4.2. Triple Set

To describe users' hierarchically extended preferences based on the knowledge graph, we recursively defined the set of n-hop relevant entities for user  $u$  as follows:

$$E_u^n = \left\{ t \mid (h, r, t) \in G \text{ and } h \in E_u^{n-1} \right\}, n = 1, 2, \dots, H \quad (4)$$

where  $E_u^0$  represents the entities contained in the news titles that the user has clicked on in the past.

We then defined the n-hop triple set of user  $u$  as follows:

$$S_u^n = \left\{ (h, r, t) \mid (h, r, t) \in G \text{ and } h \in E_u^{n-1} \right\}, n = 1, 2, \dots, H \quad (5)$$

where  $S_u^n$  are triples associated with the entities in  $E_u^n$ . There were many entities in the historical behavior of users, and these entities and their associated entities formed a large semantic network.

#### 4.3. The doc2vec Model

There are two versions of this model: the Paragraph Vector with Distributed Memory (DM) and the Distributed Bag of Words (DBOW). The DM model uses context words and a document vector to predict the target word within a context window. The DBOW model uses the document vector to predict words within a context window in the document. Despite DBOW being a simpler model, it has been shown to perform better.

The doc2vec DBOW [37] architecture is very similar to the word2vec skip-gram model, which uses the context word to predict the surrounding words in the context window. The only difference is that DBOW swaps the context word for the document vector, which is then used to predict the surrounding words in the context window. This similarity allows for the training of the two to be interleaved, thus simultaneously learning document and word vectors that are jointly embedded.

#### 4.4. Self-Attention Mechanism

The essence of an attention mechanism is: for a given target, by generating a weight coefficient to weight the input sum, the features of the input that are important to the target and those features that are not important will be identified.

The self-attention mechanism was first proposed by the Google team [38] in 2017 and applied to the Transformer language model. It is more about the connections within the input than the attention mechanism; the difference is that Q, K, and V come from the same data source (that is, Q, K, and V come from the same matrix via different linear transformations). For the text matrix, the self-attention mechanism can be used to realize the "mutual attention" of each word in the text; that is, the attention weight matrix is generated between words, and then the value-weighted summation generates a new text matrix by integrating self-attention.

### 5. Our Approach

In this section, we first introduce the overall framework of NRTEH (as illustrated in Figure 2) then discuss the process of each module with encoders. NRTEH contained three parts: the news encoder, the user encoder and the click predictor. For each news item, we

extracted a news-representation vector through the news encoder, which used two modules to extract features of the news, thereby allowing us to obtain embedding vectors set for a user’s clicked news. In the user encoder, we learned the user’s historical preferences, then aggregated the user’s topic interests and propagated the entities contained in the news that the user clicked on in the knowledge graph to obtain the user’s final representation. In the click predictor, we used the scoring function to calculate the probability of a user clicking the candidate news.

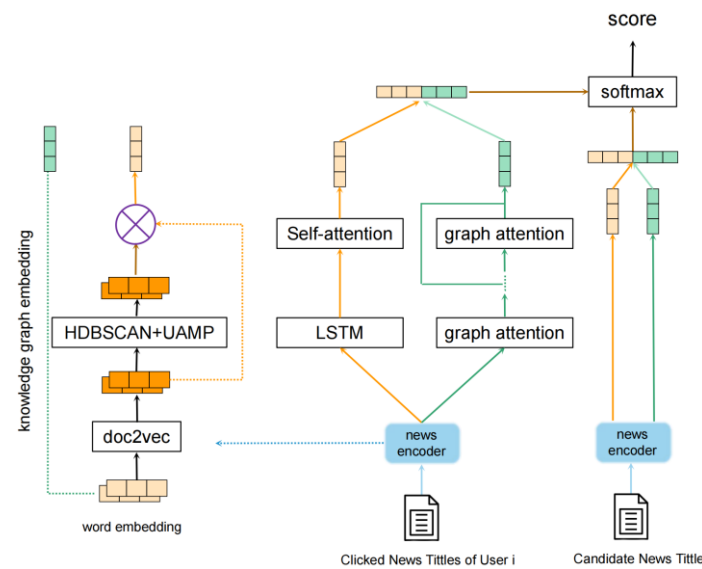


Figure 2. The framework of our NRTEH approach.

### 5.1. News Encoder

The news encoder was used to learn news representations from news titles. It contained the three modules described below.

#### 5.1.1. Word-Embedding Module

Each news title was composed of a sequence of words ( $t = [w_1, w_2, \dots]$ ). In order to construct the semantic space [39], we used the word2vec model to pretrain a matrix  $W_{n,m}$  for word vectors and a matrix  $W'_{n,m}$  for context word vectors, where n is the number of words in all news titles and m is the size of the vectors to be learned for each word.

#### 5.1.2. Knowledge-Graph Embedding Module

Each word w in a news title could be associated with an entity e in the knowledge graph. Based on these identified entities, we constructed a sub-graph and extracted all relations among them from the original knowledge graph. Given the extracted knowledge graph, we used TransE to represent the entities and relations. Through this module, the news title was converted into the vector sequence  $[e_1, e_2, \dots, e_M]$ , where M is the number of entities contained in it. We took the average value k as the knowledge-graph embedding of the title.

#### 5.1.3. Topic-Level Embedding Module

The topic information contained in news titles is vital to learning news presentation. In addition to containing explicit topics, a news title often contains multiple latent topics. We used doc2vec DBOW to determine the jointly embedded news title and word vectors. The doc2vec DBOW model consisted of a matrix  $D_{c,m}$ , where c is the number of all news titles and m is the size of the vectors to be learned for each news title. Each row of  $D_{c,m}$  contained a news title vector  $\vec{d} \in \mathbb{R}^m$ . The model also required a context word matrix  $W'_{n,m}$ , which was pretrained in the word-embedding module. For each news title d in the

corpus, the context vector  $\vec{w}_c \in W'_{n,m}$  of each word in the news title was used to predict the news title's vector  $\vec{d} \in D_{c,m}$ . The prediction was  $\text{softmax}(\vec{w}_c \cdot D_{c,m})$ , which generated a probability distribution over the corpus for each news title that contained the word.

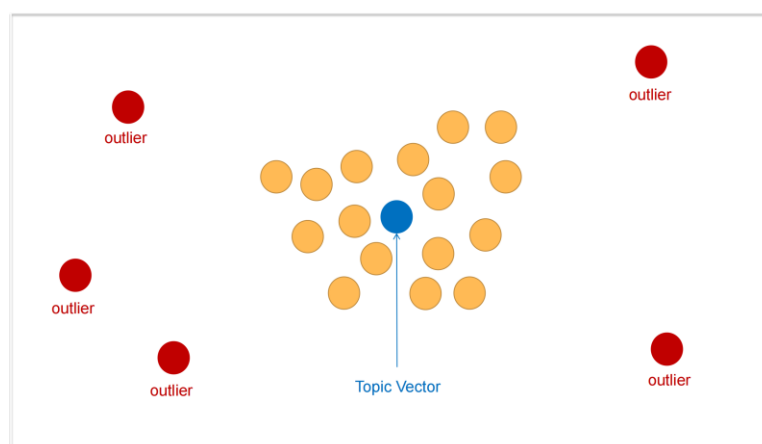
In the learning process, we used back propagation and stochastic gradient descent to update each news title vector  $\vec{d}$  in  $D_{c,m}$  and  $\vec{w}_c$  from  $W'_{n,m}$  so that the probability of the news title given the word ( $P(\vec{d} | \vec{w})$ ) was the greatest in the probability distribution over the corpus of news titles. Meanwhile, the news title vectors were required to be close to the word vector of the words in them and far from the word vector of the words not in them. This process can be regarded as each word attracting news titles that were similar to them and repelling news titles that were dissimilar to them. This resulted in a semantic space in which the news titles were closest to the words that best described them and far from words that were dissimilar to them. In this space, similar news titles were clustered around similar words in the same region, and dissimilar news titles were far apart due to clustering around the dissimilar words in different regions.

In the semantic space, an area where the news titles were highly concentrated meant that the news titles in this area were highly similar. This dense area of news titles indicated that these news titles shared one or more common latent topics. We assumed that the number of dense areas was equal to the number of topics. This method of discretizing topic allowed us to find a topic for each set of news titles that shared a topic.

We used Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [40–42] on news title vectors to find dense areas of news titles in the semantic space. However, in the 300-dimensional semantic embedding space, the news title vectors were so sparse that it was difficult to find a dense news title vector cluster, and the computational cost was high [43]. Therefore, we used the Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) [44,45] to reduce the dimension of the news title vector. In the dimension-reduced space, HDBSCAN could then be used to find dense clusters of news titles.

HDBSCAN identified the dense clusters of news titles and noise news titles in the UMAP-reduced dimension and used a noise label or a label of dense clusters to mark each news title in the semantic embedding space.

The topic vectors could be calculated by assigning labels to each dense news title cluster in the semantic embedding space. Since the news title vectors represented the topics of the news titles, the centroid or average of those vectors was the topic vector that was the most representative of the dense area of news titles from which it was calculated. The words closest to this topic vector were the words that best described it semantically. Our method was to calculate the centroid; i.e., the arithmetic means of all news title vectors in the same dense cluster. Figure 3 shows a visual example of a topic vector being calculated from a dense area of news titles.



**Figure 3.** The topic vector was the centroid of the dense area of news titles identified by HDBSCAN.

Finally, we obtained a matrix  $C_{x,m}$ , where  $x$  is the number of topics and  $m$  is the dimension of the topic vector. For each news title  $t$ , we obtained its topic embedding as follows:

$$W_t = \text{softmax}(tC^T) \quad (6)$$

$$t = W_t C \quad (7)$$

where  $W_t$  is a weight matrix of topics and  $t$  is the news title's topic embedding.

The final representation of a news title was the concat of the averaged entity embeddings and topic embedding, which was formulated as:

$$r = t \oplus k \quad (8)$$

## 5.2. User-Encoder Module

The user-encoder module was used to learn the representations of users from their browsed news. It contained two modules.

### 5.2.1. Topic-Preference-Learning Module

Usually, news browsed by the same user contains rich topics for which there is some kind of correlation between them. At the same time, we believe that users had different degrees of attention to news clicked at different moments. The purpose of this module was to learn long-term and short-term user topic preferences. Since users had different degrees of interest in each historical click news title, and the attention mechanism could capture the topics that the users were interested in, LSTM combined with the self-attention mechanism could be used to mine users' topic preferences according to the users' historical click behavior.

Using the news encoder, we obtained the news' topic embedding  $T^{c \times m}$ . Given the user's click historical matrix  $Y \in T^{c \times m}$ , we could obtain the query, key, and value in the self-attention mechanism via the nonlinear transformation of the historical behavior matrix  $Y$  as follows:

$$Q = \text{ReLU}(YW_Q) \quad (9)$$

$$K = \text{ReLU}(YW_K) \quad (10)$$

where  $W_Q \in T^{m \times m}$  and  $W_K \in T^{m \times m}$  are weight matrices of the query and key, respectively. Then, the weight matrix  $P$  could be obtained as follows:

$$P = \text{softmax}\left(\frac{QK^T}{\sqrt{m}}\right) \quad (11)$$

where  $P$  is a similar matrix of  $S$  historical behaviors. Finally, the output of self-attention could be obtained by multiplying the similarity matrix  $P$  and the history matrix  $Y$ :

$$a = PY \quad (12)$$

where  $a \in T^{S \times m}$  is the user preferences. We averaged the self-attention results to determine a single attention value:

$$p = \frac{1}{S} \sum_{i=1}^S a_i \quad (13)$$

where  $p$  is the user-topic-preference embedding.

### 5.2.2. KG-Level Preference-Propagation Module

News titles can contain multiple entities such as politicians, celebrities, companies, or institutions, which usually play a key role in the title. In addition, these entities are not independent from one another but can be linked to other entities through various relationships and then organized as a knowledge graph. The knowledge graph was a semantic network and a directed graph whose nodes and edges represented entities and relations



between the entities. The knowledge semantic features were obtained through knowledge-representation learning, and this module deeply mined the user entity preferences.

In the semantic network, the head entity was related to many entities through direct or indirect relationships, but the existence of relationships did not mean that users would have the same degree of interest in these entities. This module used graph attention networks to learn the semantic networks.

Given the average value  $k \in H^d$  of the entity embeddings in user-clicked news titles and the 1-hop triple set  $S_u^1$  of user  $u$ , we used an attention mechanism to learn the entities the user preferences.

$$x_i = \text{softmax}\left(k^T R_i h_i\right) = \frac{\exp(k^T R_i h_i)}{\sum_{(h,r,t) \in S_u^1} \exp(k^T R h)} \quad (14)$$

where  $r_i \in R^{d \times d}$  and  $h_i \in R^{d \times d}$  are the embeddings of relation  $r_i$  and head  $h_i$ , respectively. The  $x_i$  can be regarded as the weight indicating the user's interest in the entity  $h_i$  under the relation  $r_i$ . Users may have different degrees of interest in the same entity with different relations, so taking the relations into account when calculating the weights can better learn the user's interest in entities.

After obtaining the weights, we multiplied the tails in  $S_u^1$  by them, and the vector  $hop_1$  could be obtained via linear addition:

$$hop_1 = \sum_{(h,r,t) \in S_u^1} x_i t_i \quad (15)$$

where  $t_i \in R^{d \times d}$  represents the tails in  $S_u^1$ . Through this process, a user's preferences were transferred from their click history to the 1-hop relevant entities  $\mathcal{E}_u^1$  along the links in  $S_u^1$ .

By replacing  $k$  with  $hop_1$  in Equation (14), the module iterated this procedure on user  $u$ 's triple set  $S_u^i$  for  $i = 1, \dots, N$ . Therefore, a user's preference was propagated  $N$  times along the triple set from their click history,  $N$  different preference sequences were generated:  $hop_1, hop_2, \dots, hop_N$ . To represent the user's final entity-preference embeddings, we merged all embeddings:

$$f = \sum_{i=1}^N hop_i \quad (16)$$

where the embedding  $f$  is the output of this module.

The final user representation was the concat of the entity-preference embedding and topic-preference embedding, which was formulated as:

$$u = p \oplus f \quad (17)$$

### 5.3. Click Predictor

The click predictor was used to predict the probability of a user clicking a candidate news item. We denoted the representation of a candidate news  $t$  as  $r$ , and the click probability score  $\hat{y}$  was computed as follows:

$$\hat{y} = \sigma\left(u^T r\right) \quad (18)$$

where  $\sigma(x) = \frac{1}{1+\exp(-x)}$  is the sigmoid function.

## 6. Experiments

In this section of the study, we tested and evaluated our model's performance on the Bing News dataset in detail and compared it with other models. We also discussed the tuning of hyperparameters.

### 6.1. Datasets

We used the Bing News server logs from 16 May 2017 to 11 January 2018 as our dataset. It contained more than 100,000 users and their click behaviors on more than 500,000 English news articles. Each impression in the dataset contained a timestamp, a news ID, a title, and a category label. In addition, we searched for all entities that occurred in the dataset as well as entities that were within a hop of the Microsoft Satori knowledge graph and extracted all edges (triples) within them with a confidence degree greater than 0.9. The basic statistics and distribution of the news dataset are shown in Table 1.

**Table 1.** Dataset Statistics.

<b># users</b>	132,747	<b>avg. # words per title</b>	10.34
<b># news</b>	511,726	<b>avg. # entities per title</b>	3.8
<b># impressions</b>	1,116,589	<b>#triples</b>	7,558,695

### 6.2. Experiment Setup

In our experiments, we divided the dataset into a training set, a validation set, and a test set in a 6:2:2 ratio. The word embeddings were 300-dimensional and initialized by the Word2vec model. The entity embeddings were 50-dimensional and initialized via TransE. In addition, we set the hop number  $H = 2$  because a larger  $H$  value not only did not improve the performance of the model, but also increased the computational cost. These hyperparameters were tuned on the validation set. In addition, the experiment was independently repeated 10 times, and the average results in terms of the AUC and ACC was used for the performance analysis.

### 6.3. Baselines

We used the following models as baselines in our experiments:

- LSTUR [1] determined the comprehensive representation of news through the news encoder. In the user encoder, LSTUR determined the short-term representation of the user from the user's recent news clicks through the GRU network.
- LibFM [46] is a feature-based factorization model. In this paper, we took the TF-IDF features and average entity embeddings of each news item as the input feature of LibFM. In addition, we concatenated the feature of users and candidate news to feed into LibFM.
- DSSM [12] is a deep structured semantic model that uses word hashing and multiple fully connected layers to sort documents. We used the user's clicked news as the query and the candidate news as the documents.
- DeepWide [15] is a deep model for recommendation that combines a (deep) non-linear channel with a (wide) linear channel. We used the same input as for LibFM to feed both channels.
- DeepFM [13] is also a deep model for recommendation that combines a component of factorization machines and a component of deep neural networks that share the input. We used the same input as for LibFM to feed into DeepFM.
- DKN [3] is a deep knowledge-aware network for news recommendation that treats entity embedding and word embedding as multi-channel then designs a CNN model to aggregate the features together.
- RippleNet [4] is a memory-network-like approach that automatically propagates the clicked entities in the knowledge graph to capture the higher-order preferences of users.

### 6.4. Results

The results of all methods in CTR prediction are presented in Table 2.

**Table 2.** The results for AUC and accuracy in CTR prediction.

Model	LSTUR	LibFM	DSSM	DeepFM	DeepWide	DKN	RippleNet	NRTEH
AUC	0.643	0.590	0.635	0.601	0.619	0.653	0.678	<b>0.704</b>
ACC	0.604	0.554	0.606	0.574	0.567	0.607	0.645	<b>0.678</b>

The experimental results showed that our recommendation system performed the best compared with the other recommendation models. Specifically, NRTEH outperformed the baselines by 2.6% to 11.4% on AUC and 3.3% to 12.4% on ACC.

We also evaluated the influence of the maximal hop number  $H$  on NRTEH performance. The results (Table 3) showed that the best performance was achieved when  $H$  was 2 or 3. This was because if  $H$  was too small, it was difficult to explore the connection and long-distance dependence between entities; while if  $H$  was too large, it would introduce much more noise than useful signals.

**Table 3.** The results of AUC w.r.t. different hop numbers.

Hop Number	1	2	3	4	5
AUC	0.692	0.701	0.704	0.687	0.673

### 6.5. Ablation Study

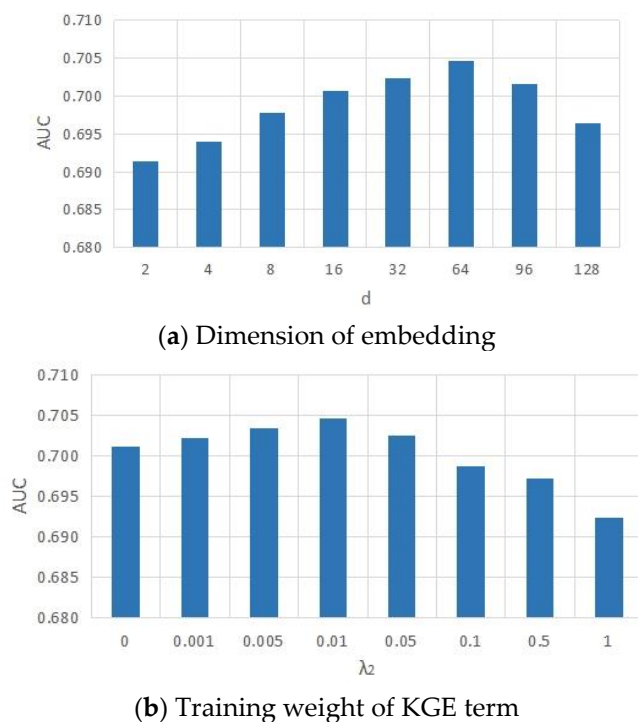
To verify the validity of our idea that attention mechanisms could improve recommendation performance, we designed an ablation study to evaluate our model. In this section of the study, instead of using attention mechanisms to capture user preferences for topics and entities, the ablation model simply aggregated them together. The experimental results are shown in Table 4. Based on these results, we found that the self-attention and graph attention were very useful. This was because users had different interests in different topics and entities, and capturing users' preferences was important for recommendations.

**Table 4.** Effectiveness of different attention networks.

Models	Bing News	
	AUC	ACC
Without attention	0.587	0.569
With self-attention	0.656	0.628
With graph attention	0.689	0.653
Both	0.704	0.678

### 6.6. Parameter Sensitivity

In this part of the experiment, we studied the effects of parameters  $d$  and  $\lambda_2$  on the model's performance. We varied  $d$  from 2 to 128 and  $\lambda_2$  from 0.0 to 1.0 while keeping the other parameters constant. The results for the AUC are shown in Figure 4. It can be seen in Figure 4a that the performance of the model improved at the beginning with an increasing  $d$  because larger dimensional embeddings could encode more useful information, but then degraded after  $d = 64$  due to possible overfitting. In Figure 4b, it can be seen that performance of NRTEH reached the best when  $\lambda_2 = 0.01$ .



**Figure 4.** Parameter sensitivities of NRTEH.

## 7. Conclusions

In this paper, we proposed NRTEH, an end-to-end framework that naturally incorporated the topic embedding and knowledge graph into a news-recommendation system. NRTEH overcame the limitations of the existing recommendation methods by addressing two major challenges in news recommendation: (1) explicit and latent topic features were extracted from news titles via topic embedding and mining users' long-term and short-term preferences for them; and (2) using a KG-level preference propagation module, it automatically propagated the users' potential preferences and explored their hierarchical interests in the KG. We conducted a great deal of experiments in a recommendation scenario. The results showed that NRTEH had a significant advantage over the strong baseline.

**Author Contributions:** Conceptualization, H.Z. and Z.S.; methodology, H.Z.; software, H.Z.; validation, H.Z. and Z.S.; formal analysis, H.Z. and Z.S.; investigation, H.Z. and Z.S.; resources, Z.S.; data curation, H.Z.; Writing—original draft, H.Z.; Writing—review & editing, Z.S.; visualization, H.Z.; supervision, Z.S.; project administration, Z.S.; funding acquisition, Z.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Key R&D projects in Hubei Province under Grant Nos. 2022BAA041 and 2021BCA124, the Open Foundation of Engineering Research Center of Cyberspace under Grant No. KJAQ202112002, and the National Key R&D Program of China under Grant Nos. 2018YFC1604000 and 2018YFC1604002.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** All data included in this study are available upon request by contacting the corresponding author.

**Conflicts of Interest:** We declare that we do not have any commercial or associative interest that represent a conflict of interest in connection with the work submitted.

## References

1. An, M.; Wu, F.; Wu, C.; Zhang, K.; Liu, Z.; Xie, X. Neural news recommendation with long-and short-term user representations. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 336–345.
2. Konstan, J.A.; Miller, B.N.; Maltz, D.; Herlocker, J.L.; Gordon, L.R.; Riedl, J. GroupLens: Applying collaborative filtering to Usenet news. *Commun. ACM* **1997**, *40*, 77–87. [[CrossRef](#)]
3. Wang, H.; Zhang, F.; Xie, X.; Guo, M. DKN: Deep knowledge-aware network for news recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1835–1844.
4. Wang, H.; Zhang, F.; Wang, J.; Zhao, M.; Li, W.; Xie, X.; Guo, M. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 417–426.
5. Lian, J.; Zhang, F.; Xie, X.; Sun, G. Towards better representation learning for personalized news recommendation: A multi-channel deep fusion approach. In Proceedings of the International Joint Conferences on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 3805–3811.
6. Son, J.-W.; Kim, A.-Y.; Park, S.-B. A location-based news article recommendation with explicit localized semantic analysis. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 28 July–1 August 2013; pp. 293–302.
7. Li, L.; Wang, D.; Li, T.; Knox, D.; Padmanabhan, B. SCENE: A scalable two-stage personalized news recommendation system. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 24–28 July 2011; pp. 125–134.
8. Okura, S.; Tagami, Y.; Ono, S.; Tajima, A. Embedding-based news recommendation for millions of users. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 13–17 August 2017; pp. 1933–1942.
9. Wang, F.; Wang, Y.; Li, D.; Gu, H.; Lu, T.; Zhang, P.; Gu, N. Enhancing CTR Prediction with Context-Aware Feature Representation Learning. *arXiv* **2022**, arXiv:2204.08758.
10. Phelan, O.; McCarthy, K.; Smyth, B. Using twitter to recommend real-time topical news. In Proceedings of the Third ACM Conference on Recommender Systems, New York, NY, USA, 23–25 October 2009; pp. 385–388.
11. Bansal, T.; Das, M.; Bhattacharyya, C. Content driven user profiling for comment-worthy recommendations of news and blog articles. In Proceedings of the 9th ACM Conference on Recommender Systems, New York, NY, USA, 16–20 September 2015; pp. 195–202.
12. Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; Heck, L. Learning deep structured semantic models for web search using clickthrough data. In Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, New York, NY, USA, 27 October–1 November 2013; pp. 2333–2338.
13. Guo, H.; Tang, R.; Ye, Y.; Li, Z.; He, X. DeepFM: A factorization-machine based neural network for CTR prediction. *arXiv* **2017**, arXiv:1703.04247.
14. Xue, H.-J.; Dai, X.; Zhang, J.; Huang, S.; Chen, J. Deep Matrix Factorization Models for Recommender Systems. In Proceedings of the International Joint Conferences on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 3203–3209.
15. Cheng, H.-T.; Koc, L.; Harmsen, J.; Shaked, T.; Chandra, T.; Aradhye, H.; Anderson, G.; Corrado, G.; Chai, W.; Ispir, M.; et al. Wide & deep learning for recommender systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 16 September 2016; pp. 7–10.
16. Sajjadi Ghaemmaghami, S.; Salehi-Abari, A. DeepGroup: Group Recommendation with Implicit Feedback. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Queensland, Australia, 1–5 November 2021; pp. 3408–3412.
17. Elkahky, A.M.; Song, Y.; He, X. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 278–288.
18. Wang, H.; Zhang, F.; Hou, M.; Xie, X.; Guo, M.; Liu, Q. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In Proceedings of the 11th ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, 5–9 February 2018; pp. 592–600.
19. Shi, S.; Ma, W.; Wang, Z.; Zhang, M.; Fang, K.; Xu, J.; Liu, Y.; Ma, S. WG4Rec: Modeling Textual Content with Word Graph for News Recommendation. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Queensland, Australia, 1–5 November 2021; pp. 1651–1660.
20. Hoffart, J.; Suchanek, F.M.; Berberich, K.; Weikum, G. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artif. Intell.* **2013**, *194*, 28–61. [[CrossRef](#)]
21. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, Vancouver, Canada, 9–12 June 2008; pp. 1247–1250.
22. Singhal, A. Introducing the knowledge graph: Things, not strings. *Off. Google Blog* **2012**, *5*, 16.

23. Wang, X.; He, X.; Cao, Y.; Liu, M.; Chua, T.-S. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 950–958.
24. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *16*, 2787–2795.
25. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. *Proc. AAAI Conf. Artif. Intell.* **2014**, *28*, 1112–1119. [[CrossRef](#)]
26. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 1, pp. 687–696.
27. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; p. 2187.
28. Wang, H.; Wang, Y.; Lian, D.; Gao, J. A lightweight knowledge graph embedding framework for efficient inference and storage. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Queensland, Australia, 1–5 November 2021; pp. 1909–1918.
29. Dong, L.; Wei, F.; Zhou, M.; Xu, K. Question answering over freebase with multi-column convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 1, pp. 260–269.
30. Wang, J.; Wang, Z.; Zhang, D.; Yan, J. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. In Proceedings of the International Joint Conferences on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 2915–2921.
31. Jin, X.; Oh, B.; Lee, S.; Lee, D.; Lee, K.-H.; Chen, L. Learning Region Similarity over Spatial Knowledge Graphs with Hierarchical Types and Semantic Relations. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 669–678.
32. Wang, H.; Zhang, F.; Zhang, M.; Leskovec, J.; Zhao, M.; Li, W.; Wang, Z. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 968–977.
33. Duvenaud, D.K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R.P. Convolutional networks on graphs for learning molecular fingerprints. In Proceedings of the 29th Conference on Neural Information Processing Systems, Montreal, Canada, 7–12 December 2015; pp. 2224–2232.
34. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 1024–1034.
35. Niepert, M.; Ahmed, M.; Kutzkov, K. Learning convolutional neural networks for graphs. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 2016; pp. 2014–2023.
36. Liu, Y.; Li, B.; Zang, Y.; Li, A.; Yin, H. A Knowledge-aware recommender with attention-enhanced dynamic convolutional network. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Queensland, Australia, 1–5 November 2021; pp. 1079–1088.
37. Rehůřek, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 22 May 2010; pp. 45–50. Available online: <http://is.muni.cz/publication/884893/en> (accessed on 10 September 2021).
38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.
39. Griffiths, T.L.; Steyvers, M.; Tenenbaum, J.B. Topics in semantic representation. *Psychol. Rev.* **2007**, *114*, 211. [[CrossRef](#)] [[PubMed](#)]
40. Campello, R.J.; Moulavi, D.; Sander, J. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Cham, Switzerland, 2013; pp. 160–172.
41. McInnes, L.; Healy, J. Accelerated hierarchical density based clustering. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 33–42.
42. McInnes, L.; Healy, J.; Astels, S. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* **2017**, *2*, 205. [[CrossRef](#)]
43. Marimont, R.B.; Shapiro, M.B. Nearest neighbour searches and the curse of dimensionality. *IMA J. Appl. Math.* **1979**, *24*, 59–70. [[CrossRef](#)]
44. McInnes, L.; Healy, J.; Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv* **2018**, arXiv:1802.03426.
45. McInnes, L.; Healy, J.; Saul, N.; Grossberger, L. Umap: Uniform manifold approximation and projection. *J. Open Source Softw.* **2018**, *3*, 861. [[CrossRef](#)]
46. Rendle, S. Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol. (TIST)* **2012**, *3*, 57. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.