LANCER : A Lifetime-Aware News Recommender System

Hong-Kyun Bae,[1] Jeewon Ahn,[1] Dongwon Lee,[2] Sang-Wook Kim[*1]

[1] Department of Computer Science, Hanyang University, South Korea
[2] College of Information Sciences and Technology, The Pennsylvania State University, USA
{hongkyun, dkswldnjs, wook}@hanyang.ac.kr, dongwon@psu.edu

Abstract

From the observation that users reading news tend to *not click outdated news*, we propose the notion of 'lifetime' of news, with two hypotheses: (i) news has a shorter lifetime, compared to other types of items such as movies or e-commerce products; (ii) news only competes with other news whose lifetimes have not ended, and which has an overlapping lifetime (*i.e.*, *limited competitions*). By further developing the characteristics of the lifetime of news, then we prenset a novel approach for news recommendation, namely, *Lifetime-Aware News reCommEndeR System* (*LANCER*) that carefully exploits the lifetime of news during training and recommendation. Using real-world news datasets (*e.g.*, Adressa and MIND), we successfully demonstrate that state-of-the-art news recommendation models can get significantly benefited by integrating the notion of lifetime and *LANCER*, by up to about 40% increases in recommendation accuracy.

Introduction

In this work, we start our investigation based on recent observation such that unlike popular domains of entertainment (*e.g.*, Netflix) and e-commerce (*e.g.*, Amazon), users in news domain *rarely click outdated news* (Wang et al. 2018; Wu et al. 2020). For instance, both (Wang et al. 2018) and (Wu et al. 2020) reported that about 85% of all news in the MIND dataset (Wu et al. 2020) had been last clicked within 48 hours from their publish times. Although there could be exceptions to this access pattern in news domain (*e.g.*, some Christmas news is seasonally popular for many years), we posit that *exploiting this peculiar access pattern in news domain, in addition to collaborative filtering and content-based modeling, could improve the accuracy of news recommendation significantly*.

We first start with two hypotheses as follows:

- H_1: News has a **lifetime**, the duration from the birth (*i.e.*, initial publish time) to the death (*i.e.*, last clicked time), which is relatively short (*i.e.*, hours, not weeks or months).
- H_2: For getting clicked from a user, news only competes with other **live** news, whose lifetime has not ended, and which has an overlapping lifetime with the news–*i.e.*, **limited competition**.

Further, we claim that existing recommendation methods (Hu et al. 2020; Liu et al. 2020; Shi et al. 2021; Tian et al. 2021; Wu et al. 2019c; An et al. 2019; Wu et al. 2019b; Mao, Zeng, and Wong 2021) *do not consider this notion of lifetime* of news. In this paper, therefore, by considering the characteristics of lifetime in a news domain, we propose a novel approach to news recommendation, named as *Lifetime-Aware News reCommEndeR system* (*LANCER*), with three key ideas below.

Idea 1: Consideration of news in competition. Based on the lifetime of news, we determine that news clicked by a user (*i.e.*, positive news) is more preferred than other non-clicked news (*i.e.*, negative news) with "overlapping" lifetimes (*i.e.*, *limited competitions*).

Idea 2: Confidence-based negative sampling among competing news. Among a user's non-clicked news with overlapping lifetimes to positive news, we find truly negative news by estimating the *confidence* based on their popularity. For instance, we assume that when less-popular news is not clicked, it is more likely to be truly negative since a user probably did not like it.

Idea 3: Consideration of remaining lifetime of news. To curb recommending news whose lifetime has ended or is near death, we adjust the predicted preference scores for news by considering the amount of their remaining lifetime. Via this adjustment, we recommend news with both highly-predicted preferences and sufficiently-remaining lifetimes (*i.e.*, preferred and relatively young news).

As the notion of news lifetime is orthogonal to recommendation kernels, *LANCER* can be independently applied to existing news recommendation models (*e.g.*, NRMS (Wu et al. 2019c), LSTUR (An et al. 2019), NAML (Wu et al. 2019b), and CNE-SUE (Mao, Zeng, and Wong 2021)). In the Evaluation section, we successfully demonstrate the value of *LANCER* by showing that several state-of-the-art news recommendation algorithms get significantly benefited by incorporating *LANCER*. Our main contributions are as follows:

- **Observation**: We formulate the notion of *lifetime* in news recommendation by identifying the period during which the majority of clicks occur for news, and quan-

[*]Corresponding author.

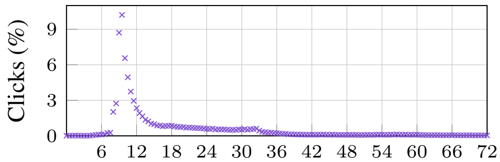


Figure 1: Distribution of average click ratios for news by users over time (Adressa).

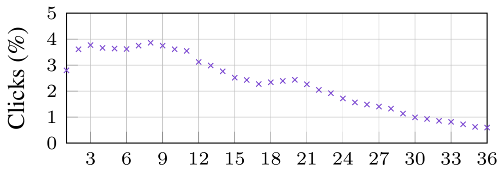


Figure 2: Distribution of average click ratios for items by users over time (Netflix).

titatively present the length of the average lifetimes of news.

- **Claim**: We propose a new concept of *limited competitions* between news, and claim for the first time that the models for news recommendation can be benefited when trained based on these competitions.
- **Approach**: We propose a novel approach, *LANCER*, consisting of the three aforementioned key ideas.
- **Evaluation**: We demonstrate that *LANCER* can significantly enhance the accuracy of the existing models for news recommendation.

Motivation

The period that news is clicked intensively by users tends to be *limited*, unlike the other domains such as *Over-The-Top media* (OTT) or e-commerce. To verify this tendency, we analyzed both datasets of news from Adressa (Gulla et al. 2017) and movies/dramas from Netflix[1] by examining the distribution of average click ratios per item over time with the following equation:

$$y = \Sigma_{d \in D}\left(\frac{c(d,t)}{C(d)}\right)/|D| \times 100\ (\%) \quad (1)$$

where $c(d,t)$ indicates the number of clicks that an item d received from all users at the time after t from its publish time; $C(d)$ indicates the total number of clicks that d has received during the entire period. Note that we computed the click ratio, not the number of clicks, to reduce the tendency to be biased towards some (popular) items that received a very high number of clicks from users. For the Netflix dataset, we regarded the *first click time* that an item received as its publish time, since the publish times of movies/dramas from Netflix are not publicly available.

[1]https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data

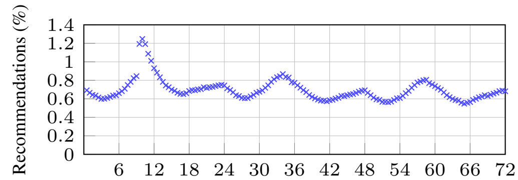


Figure 3: Percentage of news recommended by NRMS (Wu et al. 2019c) by elapsed time from the publication time (Adressa).

The results are illustrated in Figures 1 and 2 for the Adressa and Netflix, respectively. In Figure 1, news on Adressa receives clicks from users intensively until 6–18 hours, but very few clicks occur after 48 hours from their publish times. On the other hand, in Figure 2, items on Netflix strongly tend to receive more than a certain number of clicks from users for *nearly unlimited long periods*. Notably, on average, we observed that it could take up to *32 months* for an item on Netflix to receive about 80% of all clicks from users, while only for *36 hours* in the news domain, which is extremely short compared with Netflix. In other words, it supports a hypothesis, H_1 that news has a relatively shorter lifetime than items such as movies or e-commerce products.

Definition 1 (Lifetime (m)) *The period from the initial publish time to the last clicked time, where m% of clicks occur. For instance, when we empirically set m=80 for a news domain, we observe that lifetime(80)=36 hours.*

Various studies based on *deep learning* (DL) models (Hu et al. 2020; Liu et al. 2020; Shi et al. 2021; Tian et al. 2021) have been popularly conducted for news recommendation. For instance, DL models such as *Attention Network* (Vaswani et al. 2017), *Convolutional Neural Network* (CNN) (Lecun et al. 1998), or *Long-Short Term Memory* (LSTM) (Hochreiter and Schmidhuber 1997) have been employed to infer the user preference for news. However, the existing news recommendations have *not considered the lifetimes of news* while training the models and recommending the news to a user: they do not take into account the *competitions among news* to infer user preference for news and do not consider the *remaining lifetime of news* at the recommendation time.

To show the limitation of the existing studies, in Figure 3, we investigate the distribution of recommendations across hours, where one of the state-of-the-art methods (*i.e.*, NRMS (Wu et al. 2019c)) still recommends a lot of news after 48 hours from their initial publish time (*i.e.*, relatively old news). That is, many recommendations in the right-hand side of Figure 3 are potentially wasted as they are unlikely to be clicked by users, as shown in the right-hand side of Figure 1. In the following section, we elaborate on our proposed approach to address these limitations.

Previous Studies

There have been a few studies that introduce the concept of a lifetime of news with their own definitions. (Wang et al.

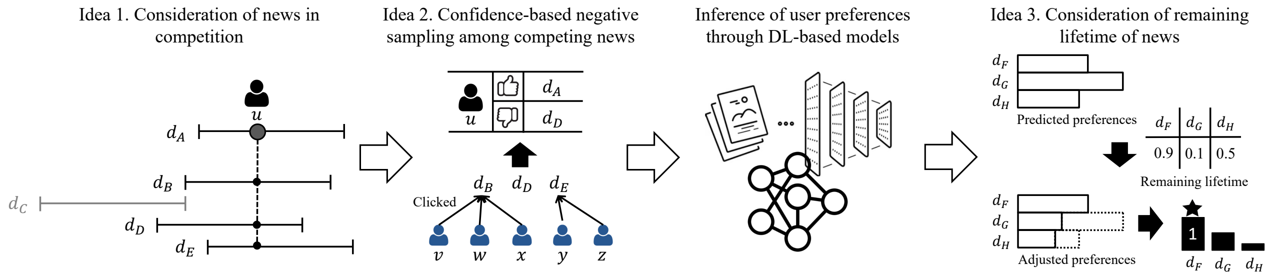


Figure 4: Overview of our proposed approach for news recommendation, *LANCER*.

2018) and (Wu et al. 2020) regarded the period from the publication to the end of the clicks for a news as its lifetime. But they *overlooked* the characteristic that news competes with only other news that has an overlapping lifetime for getting clicked by a user. (Castillo et al. 2014) and (Ni et al. 2021) tried to understand the life-cycle of the news topics. To this end, (Castillo et al. 2014) investigated the users' *social media reactions* to the news (*e.g.*, *tweets* or *tags* for the news), and (Ni et al. 2021) identified the change in the number of publications of the news with respect to specific topics over time. Although they show interesting observations, they are on a different research line from us in terms of not focusing on news recommendations.

The Proposed Approach: *LANCER*

Overview

In this section, we present how to design our *LANCER* for news recommendation with consideration of the characteristics related to lifetime in a news domain. The overall procedure in *LANCER* is described in Figure 4. In Idea 1, within a set of news with overlapping lifetimes (*i.e.*, the news *competing with each other*), we determine that a clicked (positive) news by the user (*i.e.*, d_A) is more preferred than the non-clicked (negative) news (*i.e.*, d_B, d_D, and d_E). In Idea 2, we aim to train the negative news that can be *highly confident for the negativeness* among the non-clicked news competing with the corresponding positive news of a user. We make less popular news (*i.e.*, d_D) be determined as u's negative news with higher confidence since u was not likely to select it either like other users. Next, we train existing DL-based models to predict users' preferences (*e.g.*, NRMS (Wu et al. 2019c)) through the positive/negative news determined by our Ideas 1 and 2. In Idea 3, we *adjust* the scores of the user's predicted preferences for news by considering their *remaining lifetimes at the time of recommendation*. Consequently, the news with both *highly predicted preferences and enough remaining lifetimes*, such as d_F, is recommended for u in our *LANCER* approach.

Consideration of the News in Competition (Idea 1)

In terms of the lifetime in a news domain, we made a hypothesis, H_2 that each news *competes only with the news whose lifetimes have not ended yet* for the clicks from users (*i.e.*, *limited competitions*), rather than competing with all news. Thus, the goal of Idea 1 is to find the news in competition with each other and to determine the user's positive/negative news among them.

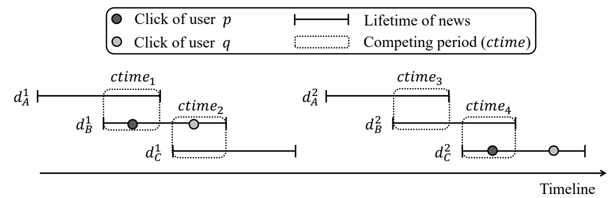


Figure 5: Clicks of two users for six news.

Finding news in competition with each other. To this end, we first leverage *36 hours*, statistically observed from analyzing a real-world dataset, as the length of lifetime of news (please refer to the previous section). Then, we consider the *news with overlapping lifetimes* to be competing with each other. For a news d_i, we formally define a set of news that have competitions with d_i, $CPT(d_i)$, as follows:

$$CPT(d_i) = \{d_j || ltime(d_i) \cap ltime(d_j)| > 0,\ d_j \in \mathcal{D}\}, \quad (2)$$

where $ltime(d_i)$ and $ltime(d_j)$ indicate the lifetimes, the periods of 36 hours since the publish times of d_i and d_j, respectively; $\mathcal{D}$ indicates a set of all news. According to Eq. 2, we determine (one or several) news which has overlapping lifetime with d_i as a set of competing news for d_i. Here, we define such an overlapping period between the lifetimes of the news (such as d_i and $d_j \in CPT(d_i)$) as the *competing period*, $ctime(d_i, d_j)$, between them (*i.e.*, $ctime(d_i, d_j) = ltime(d_i) \cap ltime(d_j)$).

Suppose that, as in Figure 5, there are two users, p and q, and six news, d_A^1, d_A^2, d_B^1, d_B^2, d_C^1, and d_C^2, that have different periods of lifetimes with each other. Here, d_A^n, d_B^n and d_C^n deal with the topics A, B, and C, respectively. The dotted boxes, $ctime_1$, $ctime_2$, $ctime_3$, and $ctime_4$, depict the competing periods between the corresponding news in competition with each other. According to Eq. 2, for each news, the sets of competing news are determined as follows: $CPT(d_A^1) = \{d_B^1\}$; $CPT(d_B^1) = \{d_A^1, d_C^1\}$; $CPT(d_C^1) = \{d_B^1\}$; $CPT(d_A^2) = \{d_B^2\}$; $CPT(d_B^2) = \{d_A^2, d_C^2\}$; $CPT(d_C^2) = \{d_B^2\}$.

Determining the positive/negative news. We identify the user's positive/negative news among the ones in competitions with each other. Specifically, first, we regard a news clicked by a user as her positive item. Then, the news, that she did not click *during the competing period with the corresponding positive news*, are determined as her negative item. Only these pairs of positive/negative news determined by Idea 1 are engaged for training the recommendation models.

For example, in Figure 5, during $ctime_1$, the user p clicked only d_B^1. So, d_B^1 and d_A^1 are regarded as p's positive and negative news, respectively. Then during $ctime_2$, since the user q clicked only d_B^1, q's positive and negative news are considered as d_B^1 and d_C^1, respectively. On the other hand, during $ctime_3$, since no news were clicked by the users, we are not able to distinguish which one is positive or negative between d_A^2 and d_B^2. During $ctime_4$, d_C^2 is regarded to be more preferred than d_B^2 by the user p. Consequently, in our approach, the following positive/negative news for each user are engaged to train models for news recommendations: d_B^1/d_A^1 and d_C^2/d_B^2 for p; d_B^1/d_C^1 for q.

Limitations of the existing studies. However, the existing studies have *not considered the notion of limited competitions among the news* in determining the user's positive/negative items due to their inconsideration of the characteristics of a lifetime in a news domain (Hu et al. 2020; Liu et al. 2020; Shi et al. 2021; Tian et al. 2021). Specifically, any clicked/non-clicked news by a user are considered as positive/negative items of the user, respectively. Thus, even the non-clicked news that have *never competed* with the clicked news can be trained *wrongly* as negative for the corresponding clicked (positive) news.

In Figure 5, through the positive/negative news determined in our *LANCER* approach (*i.e.*, d_B^1/d_A^1 and d_C^2/d_B^2 for p; d_B^1/d_C^1 for q), the *correct* orders of topic preferences of each user can be figured out as follows: $B > A$ and $C > B$ for p, thus $C > B > A$ for p; $B > C$ for q. On the other hand, in the existing studies, the order of $B = C > A$ is determined for both users p and q equally, because they both clicked d_B^1 and d_C^2 but not d_A^n. In terms of inferring user preferences, this *wrong* ordering can cause the following problems. First, two users p and q can be trained (wrongly) to have *similar* tastes for the topics. But their tastes are *not similar* with each other in reality since, for the topics B and C, p prefers C over B, whereas q prefers B over C. Second, q's preference for topic A can be trained (wrongly) to be *negative*. But the reason q did not click the news d_A^n was not because q did not prefer it, but because q could not meet the news. It is thus more adequate to regard q's preference for topic A as *unknown*, not negative. Consequently, the user preferences can be inferred incorrectly in the existing methods, which can lead to low accuracy in recommendations. It will be empirically validated in the Evaluation section.

Confidence-based Negative Sampling among Competing News (Idea 2)

For a user's positive news, in general, there are many competing non-clicked ones that can be regarded as negative (*i.e.*, a user's non-clicked news with a lifetime overlapping with the positive news). But among them, many news may have not been clicked because the user was unaware of their existence, rather than having negative preferences for them. Thus, the goal of Idea 2 is to sample the non-clicked news that can be *confident to be the user's negative ones along with the corresponding positive news*, which will be more beneficial to training the models.

Defining a confidence. To this end, we determine *confidence in the negativeness* of each non-clicked news by the user, depending on its *popularity*. It is assumed that her non-clicked news with a *less popularity* may have *higher confidence in the negativeness* since she probably did not like it either. To estimate this *popularity-based confidence* for her negative news, we investigate the number of (other) users who had clicked the negative news, when she clicked the positive news as a result of the competition with that negative news. It can be formally defined as follows:

$$conf(u, d_i, d_j) = 1 - \left(\frac{log(pop(u, d_i, d_j))}{\sum_{d_k \in CPT(d_i)} log(pop(u, d_i, d_k))} \right), \quad d_j \in CPT(d_i), \quad (3)$$

where d_i and d_j indicate a user u's positive and negative news, respectively; $conf(u, d_i, d_j)$ indicates a confidence in the negativeness of d_j competing with d_i by u; $pop(u, d_i, d_j)$ indicates the number of (other) users who had clicked d_j until the click time of u for d_i (*i.e.*, the number of users who had clicked d_j *after* u's click time for d_i is *not engaged* to compute $pop(u, d_i, d_j)$). Here, to alleviate too large difference in confidences of the news that may arise due to the gap of popularities, we try to smooth the news's popularities by obtaining the *logarithmic* values of them while determining the confidence.

Then, through employing the confidence determined by Eq. 3 as the negative sampling probability, we decide a user's negative news to be trained by the models along with the corresponding user's positive news. Consequently, the models are trained to predict the user's lower preferences for such negative news than a corresponding positive news.

We note that there are some recent studies focusing on *predicting the popularity* of news through the trained DL-based model such as attention networks (Wang et al. 2021; Wu, Wu, and Huang 2021). These popularity prediction methods can also be applied independently to our *LANCER* approach to determine confidence in negative news (for Idea 2). We leave it as our future work.

Training the DL-based models. In our *LANCER*, we employ the existing DL-based models that had been proposed for news recommendations (*e.g.*, NRMS (Wu et al. 2019c), CNE-SUE (Mao, Zeng, and Wong 2021)) to *represent* the users and the news by *embedding vectors* of them, since the notion of news lifetime is orthogonal to any recommendation kernels. To infer the user preferences, we determine the K negative news for a user u's positive news according to the *confidence-based sampling probability*. Then we train the model with these (K+1) news, by optimizing the following loss function:

$$\mathcal{L}_u = - \sum_{d_i \in \mathcal{I}_u} log \left(\frac{e^{\hat{p}(u,d_i)}}{e^{\hat{p}(u,d_i)} + \sum_{j=1}^{K} e^{\hat{p}(u,d_j)}} \right) \quad (4)$$

where d_i and d_j indicate a user u's positive and negative news, respectively; $\mathcal{I}_u$ indicates a set of positive news of u;

$\hat{p}(u, d_i)$ and $\hat{p}(u, d_j)$ indicate the predicted preferences of u for d_i and d_j, respectively, which is computed by *dot product* between the corresponding embedding vectors (*e.g.*, $\overrightarrow{\boldsymbol{u}}$ and $\overrightarrow{\boldsymbol{d_i}}$ for $\hat{p}(u, d_i)$).

Consideration of Remaining Lifetime (Idea 3)

Although the models have been trained to achieve good quality of recommendation, a user will not be satisfied with the recommendations, if those models provide the news that *have already ended or are nearing the end of their lifetimes* to the user. Thus, the goal of Idea 3 is to recommend particularly the news that have *enough remaining lifetimes as well as the highly predicted preferences*.

Toward this end, we adjust the predicted preference for news with a consideration of its *remaining lifetimes at the time of recommendation* for a user. Specifically, we *lower* the predicted preferences for the news that have *short remaining lifetimes*, currently, to decrease the probability that such news will be recommended for users. Here, we employ the *sigmoid function* to decide how much to decrease the predicted preferences of the news according to the lengths of their remaining lifetimes, which can be formally defined as follows:

$$\hat{p}(u, d_i, t_{rec}) = \frac{1}{1 + e^{-\alpha \cdot |rtime(d_i, t_{rec})|}} \cdot \hat{p}(u, d_i) \qquad (5)$$

where $\hat{p}(u, d_i, t_{rec})$ indicates the adjusted preference for news d_i at the (current) time of recommendation, t_{rec}; α indicates a hyperparameter for scaling the degree to which the predicted preference of the news is lowered, according to its length of remaining lifetime; $|rtime(d_i, t_{rec})|$ indicates the length of remaining lifetime of d_i at t_{rec}, which is determined by $|ltime(d_i)| - |(t_{rec} - t_{pub(d_i)})|$, where $t_{pub(d_i)}$ denotes the publish time of d_i. Through this proposed adjusting scheme in our *LANCER* approach, we can enforce the news with both *highly predicted preferences and enough remaining lifetimes* be mainly recommended to users.

Discussion

In our approach, we are not showing a *brand new model* based on DL techniques. Rather, we figure out properly the domain characteristics (which existing studies have not considered yet, thus overlooking) through the careful analysis of real-world datasets. Then, we propose a novel approach to effective DL-based news recommendations based on these characteristics, which is an *important contribution in the field of typical data science*.

We also stress that there are existing studies for determining negative news by using the information of the news's *impressions* for users (Wu et al. 2019c; An et al. 2019; Wu et al. 2019b; Mao, Zeng, and Wong 2021; Wu et al. 2019a; Qi et al. 2021b; Wu, Wu, and Huang 2021; Wang et al. 2021; Qi et al. 2021a; Wu et al. 2021b). In these methods, the news that were not clicked in the same *impression log* with the user's positive news are considered as negative. Here, the news in the impression log for a user are identical to the news recommended (by online platform) for the user (Wu et al. 2020, 2021a): they are already *close to the user's taste*. Therefore, training the non-clicked news in the impression log as negative items for a user is the same as training only *hard negative* news, along with the corresponding positive news.[2] In the following section, we demonstrate that the models trained by such impression log are also *less effective* in finding users' favorable news than the models trained in our *LANCER* approach.

Datasets	# of users	# of items	# of clicks	Sparsity
Adressa	259,709	24,060	6,067,109	99.9%
MIND	200,000	78,316	4,627,681	99.7%

Table 1: Statistics of two real-world datasets

Empirical Evaluation

Experimental Setup

Datasets. We conduct experiments on two popular real-world datasets: MIND (Wu et al. 2020) and Adressa (Gulla et al. 2017) as shown in Table 1. We note most of existing works use either MIND (Wu et al. 2019a; Qi et al. 2021b; Wu, Wu, and Huang 2021; Wang et al. 2021; Qi et al. 2021a; Wu et al. 2021b) or Adressa (Hu et al. 2020; Liu et al. 2020; Shi et al. 2021; Tian et al. 2021), in their evaluation. We note that the publication time of news and the user's click time for news are not available on MIND. So, for our research, we regarded the time when news was first impressed for any user as its publication time, and the time when news clicked by the user was impressed for her as her click time for the news on MIND.

Specifics for evaluation. While training news recommendation models, we employed 8 as the value of K in Eq. 4. Then, to evaluate the accuracy of news recommendation, we constructed test sets to have 20 negative news for a user's single positive news during the test period (*i.e.*, 7-th day and 5-th week in MIND and Adressa, respectively). Here, we sampled only from the non-clicked news competing with positive news in order to evaluate the accuracy for *live news*. For computing the accuracy, we used three popular metrics, *AUC*, *MRR*, and *NDCG* (namely, G), as the existing studies (Wu et al. 2019c; An et al. 2019; Wu et al. 2019b; Mao, Zeng, and Wong 2021). Moreover, we conducted t-tests with a 95% confidence level to verify the difference in accuracies between the *base models* and the models equipped with our *LANCER*. As the base models, we employed the following four state-of-the-art models: NRMS (Wu et al. 2019c); LSTUR (An et al. 2019); NAML (Wu et al. 2019b); and CNE-SUE (Mao, Zeng, and Wong 2021).

Experimental Results

Our experiments are designed to answer the following four key evaluation questions (EQs).

- **EQ1.** How effective is it to determine a user's negative news by considering the limited competitions?

[2] A *hard negative item* means a negative item that is easily predicted *incorrectly* as a positive one by the model. That is, it is a negative item that has many *factors that users may prefer* (Hariharan, Malik, and Ramanan 2012).

Metric	NRMS					LSTUR					NAML					CNE-SUE				
	Orig* (O)	*LANCER*			Gain (%)	Orig** (O)	*LANCER*			Gain (%)	Orig** (O)	*LANCER*			Gain (%)	Orig*** (O)	*LANCER*			Gain (%)
		C**	*C/N***	*C/N/R*	vs. O		*C****	*C/N***	*C/N/R*	vs. O		*C***	*C/N**	*C/N/R*	vs. O		*C***	*C/N***	*C/N/R*	vs. O
AUC	.551	.620	.637	**.663**	20.3	.571	.587	.603	**.617**	8.1	.632	.687	.715	**.740**	17.1	.600	.638	.644	**.657**	9.5
MRR	.225	.255	.258	**.280**	24.4	.216	.223	.228	**.248**	14.8	.263	.270	.296	**.344**	30.8	.226	.250	.257	**.275**	21.7
G@5	.208	.251	.250	**.280**	34.6	.196	.211	.217	**.239**	21.9	.255	.275	.312	**.362**	42.0	.219	.239	.250	**.272**	24.2
G@10	.273	.329	.333	**.363**	33.0	.283	.293	.301	**.321**	13.4	.334	.364	.392	**.435**	30.2	.295	.322	.336	**.355**	20.3

Table 2: Comparison of accuracy between the original method (*i.e.*, Orig) and variants from *LANCER* for each base model, where the Gain (%) indicates the degree of improvement achieved by *LANCER*$_{C/N/R}$ compared with Orig. $*$, $**$, and $***$ denote $p < 0.05$, $p < 0.005$, and $p < 0.0005$ for the paired t-test with *LANCER*$_{C/N/R}$, respectively (Adressa)

Metric	NRMS					LSTUR					NAML					CNE-SUE				
	Imp** (I)	Orig* (O)	*LANCER*	Gain (%)		Imp** (I)	Orig (O)	*LANCER*	Gain (%)		Imp** (I)	Orig** (O)	*LANCER*	Gain (%)		Imp* (I)	Orig* (O)	*LANCER*	Gain (%)	
			C	vs. I	vs. O			*C*	vs. I	vs. O			*C*	vs. I	vs. O			*C*	vs. I	vs. O
AUC	.688	.836	**.850**	23.5.	1.7	.587	.663	**.688**	17.2	3.8	.732	.860	**.879**	20.1	2.2	.767	.860	**.898**	17.1	4.4
MRR	.324	.520	**.526**	62.3	1.2	.223	.351	**.352**	57.8	0.3	.399	.552	**.562**	40.9	1.8	.328	.407	**.433**	32.0	6.4
G@5	.325	.560	**.567**	74.5	1.2	.201	.354	**.362**	80.1	2.3	.410	.597	**.614**	49.8	2.8	.542	.690	**.741**	36.7	7.4
G@10	.404	.604	**.614**	52.0	1.7	.287	.414	**.424**	47.7	2.4	.475	.635	**.652**	37.3	2.7	.600	.743	**.792**	32.0	6.6

Table 3: Comparison of accuracy among the impression-based method (*i.e.*, Imp), the original method (*i.e.*, Orig), and *LANCER*$_C$ for each base model. $*$ and $**$ indicate $p < 0.05$ and $p < 0.005$ for the paired t-test with *LANCER*$_C$, respectively (MIND)

- **EQ2.** How effective is it to engage the popularity-based confidence for negative sampling?
- **EQ3.** How effective is it to consider the remaining lifetimes of news along with predicted preferences?
- **EQ4.** How does the accuracy of recommendation vary according to the parameter α?

EQ1. To answer EQ1, we designed the variant *LANCER*$_C$ that samples *randomly* K non-clicked news of each user only from a set of her non-clicked news that *had competition with the corresponding positive news* (*i.e.*, Idea 1). Then, we compared it with the original method (*i.e.*, Orig), which samples randomly K non-clicked news of each user *without consideration of lifetime*, for each base model.

Table 2 reports the results in terms of the recommendation accuracy on Adressa. We can observe that any models equipped with *LANCER*$_C$ consistently outperform the original methods, regardless of the metrics. Specifically, it improves the accuracy *significantly* by up to about 20%, 15%, 10%, and 10% for NRMS, LSTUR, NAML, and CNE-SUE, respectively, where the gain is computed by (*LANCER*$_C$−Orig)/Orig×100. These consistent results verify the effectiveness of Idea 1 in our approach: it could successfully address the limitation of existing studies that do not take into account the *limited competitions among news, which is related to lifetime of news*.

Table 3 reports the results on MIND, where Imp samples randomly K non-clicked news of each user from the *same impression log* with the corresponding positive news. By comparing *LANCER*$_C$ with Orig and Imp, we can make the following observations: (i) as same as on Adressa, the models equipped with *LANCER*$_C$ outperform those with Orig, by up to about 3% and 7.5%, respectively, for NAML and CNE-SUE; (ii) the models equipped with Imp display even lower accuracy than Orig, which indicates that *training the model by negative sampling only from the impression logs is hardly effective for inferring user preference*.

Metric	*LANCER*$_{C/(1-N)}$			
	NRMS	LSTUR	NAML	CNE-SUE
AUC	0.615	0.574	0.665	0.621
MRR	0.242	0.219	0.272	0.244
G@5	0.228	0.197	0.268	0.229
G@10	0.312	0.281	0.350	0.316

Table 4: Accuracy of the variant from *LANCER*$_{C/(1-N)}$ that is contrary to Idea2 (Adressa)

EQ2. To answer EQ2, we also designed the variant *LANCER*$_{C/N}$ that samples mainly negative news with *low popularity* by giving them high probabilities (*i.e.*, integrating both Ideas 1–2). Then, we compared it with the variant *LANCER*$_C$ for each base model.

In Table 2, we observe that any models equipped with *LANCER*$_{C/N}$ universally outperform the models equipped with *LANCER*$_C$ on Adressa. The accuracies of NRMS, LSTUR, NAML, and CNE-SUE are enhanced by up to about 2.7%, 2.8%, 13.5% and 4.6%, respectively, where the gain is computed by (*LANCER*$_{C/N}$−*LANCER*$_C$)/*LANCER*$_C$ × 100. Also, Table 4 reports the results from the additional variant, *LANCER*$_{C/(1-N)}$, that samples mainly negative news with *high popularity* by giving them high probabilities (*i.e.*, contrary to Idea 2). In terms of accuracy, the order of the three variants is the same as (*LANCER*$_{C/N}$ > *LANCER*$_C$ > *LANCER*$_{C/(1-N)}$) regardless of base models. These results indicate the following observations: (i) negative sampling by confidence based on *wrong* assumption (*i.e.*, *LANCER*$_{C/(1-N)}$) can show worse accuracy than ran-

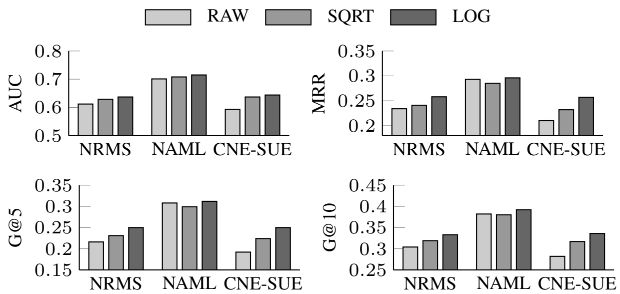


Figure 6: Comparison of recommendation accuracy with different smoothing techniques for the confidence (Adressa).

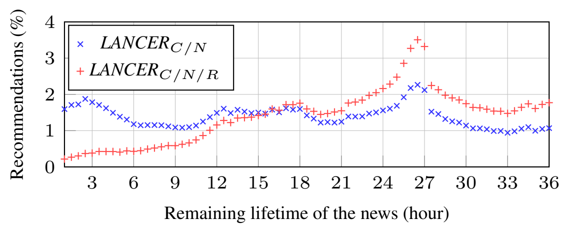


Figure 7: Percentage of news recommended by CNE-SUE (Mao, Zeng, and Wong 2021) by the remaining lifetimes of news (Adressa).

dom sampling without the consideration of confidence for the negative sampling (*i.e.*, *LANCER*$_C$); (ii) our proposed scheme for confidence-based negative sampling with the popularity of news (*i.e.*, *LANCER*$_{C/N}$) contributes to improving the accuracy of recommendation for news.

Moreover, we investigate how recommendation accuracy changes depending on the *smoothing function* used in Eq. 3 for computing confidence of negative news. To this end, we compare the result from using a log function with the results from following two variants: (**RAW**) using the popularity of news without any smoothing; (**SQRT**) using the square root value of popularity (*i.e.*, weaker smoothing than a log function). As illustrated in Figure 6, the log function for smoothing (LOG) shows the best accuracies, regardless of base models, and the results with SQRT are higher than those with RAW for most cases. These results indicate that it is necessary to use properly smoothed values when computing popularity-based confidence for news due to severe differences in popularity among news.

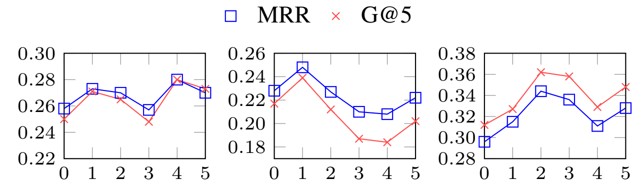
EQ3. To answer EQ3, we compare our *LANCER* of integrating all three key ideas (*i.e.*, *LANCER*$_{C/N/R}$) with the variant of *LANCER*$_{C/N}$. Here, we set α to the value showing the best accuracy of recommendation for each model, respectively (please refer to EQ4).

In Table 2, we observe that any models equipped with *LANCER*$_{C/N/R}$ consistently outperform the models equipped with *LANCER*$_{C/N}$ on Adressa. The models equipped with *LANCER*$_{C/N/R}$ improve the accuracy by up to about 16% and 9%, for NAML and CNE-SUE, respectively, compared with the models equipped with *LANCER*$_{C/N}$. Here, the gain is computed by (*LANCER*$_{C/N/R}$−*LANCER*$_{C/N}$)/*LANCER*$_{C/N}$ $\times$ 100. It demonstrates that the consideration of remaining lifetimes of news together is effective for recommendations, rather than simply considering the predicted preferences only.

In addition, we employ top-1 news for each user recommended by CNE-SUE equipped with two variants (*i.e.*, *LANCER*$_{C/N}$ and *LANCER*$_{C/N/R}$), respectively, and investigate the remaining lifetime of each corresponding news. Figure 7 shows the results, where x-axis denotes the length of remaining lifetime of the news at recommendation time and y-axis indicates the ratio of corresponding recommended news. From the figure, it can be clearly identified that more news with long remaining lifetimes (close to 36) can be recommended by *LANCER*$_{C/N/R}$ than by *LANCER*$_{C/N}$. Consequently, our proposed *LANCER*$_{C/N/R}$ integrating all our Ideas 1–3 is beneficial to recommending the news with both *highly predicted preferences and enough remaining lifetimes*.

EQ4. To answer EQ4, we show the changes of accuracy with different values for parameter α, which is used to decide the degree of adjustment in Idea 3, ranging from 0.1 to 0.5 in increment of 0.1. Smaller values of α significantly lower the predicted preferences of news with the small length of remaining lifetime.

In Figure 8, where x-axis denotes α ($\times$10) and y-axis indicates the accuracy from the corresponding metrics. Regardless of the metrics, the results with α=0.4, α=0.1, and α=0.2 show the best performances for NRMS, LSTUR, and NAML, respectively. We leveraged these respective values of α for each base model in EQ3.



(a) NRMS (b) LSTUR (c) NAML

Figure 8: Accuracies obtained by varying α (Adressa).

Conclusion

In this paper, we exploited the characteristics of *lifetime* in a news domain: such that (i) the lifetime of news is relatively shorter than that of movies or e-commerce products; and (ii) news only competes with other news whose lifetime has not ended, and which has an overlapping lifetime (*i.e.*, limited competitions). We proposed a novel approach to news recommendation, *LANCER*, with three key ideas: (i) consideration of news in competition; (ii) confidence-based negative sampling among competing news; and (iii) consideration of remaining lifetime of news. In the empirical studies using two real-world news datasets, we demonstrated that several state-of-the-art news recommendation algorithms get significantly benefited by incorporating our *LANCER*.

Acknowledgments

The work of Sang-Wook Kim was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2022-00155586 and No.2022-0-00352). The work of Dongwon Lee was supported by the NSF award #2114824.

References

An, M.; Wu, F.; Wu, C.; Zhang, K.; Liu, Z.; and Xie, X. 2019. Neural News Recommendation with Long- and Short-term User Representations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 336–345.

Castillo, C.; El-Haddad, M.; Pfeffer, J.; and Stempeck, M. 2014. Characterizing the life cycle of online news stories using social media reactions. In *ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW)*, 211–223.

Gulla, J. A.; Zhang, L.; Liu, P.; Ozgobek, O.; and Su, X. 2017. The Adressa dataset for news recommendation. In *International Conference on Web Intelligence (WI)*, 1042–1048.

Hariharan, B.; Malik, J.; and Ramanan, D. 2012. Discriminative Decorrelation for Clustering and Classification. In *European Conference on Computer Vision (ECCV)*, 459–472.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Hu, L.; Xu, S.; Li, C.; Yang, C.; Shi, C.; Duan, N.; Xie, X.; and Zhou, M. 2020. Graph Neural News Recommendation with Unsupervised Preference Disentanglement. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 4255–4264.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Liu, R.; Peng, H.; Chen, Y.; and Zhang, D. 2020. HyperNews: Simultaneous news recommendation and active-time prediction via a double-task deep neural network. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 3487–3493.

Mao, Z.; Zeng, X.; and Wong, K.-F. 2021. Neural News Recommendation with Collaborative News Encoding and Structural User Encoding. In *Findings of the Association for Computational Linguistics: EMNLP (EMNLP Findings)*, 46–55.

Ni, X.; Bu, S.; Adams, L.; and Markov, I. L. 2021. Prioritizing Original News on Facebook. In *ACM International Conference on Information and Knowledge Management (CIKM)*, 4046–4054.

Qi, T.; Wu, F.; Wu, C.; and Huang, Y. 2021a. Personalized News Recommendation with Knowledge-aware Interactive Matching. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 61–70.

Qi, T.; Wu, F.; Wu, C.; Yang, P.; Yu, Y.; Xie, X.; and Huang, Y. 2021b. HieRec: Hierarchical User Interest Modeling for Personalized News Recommendation. In *Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 5446–5456.

Shi, S.; Ma, W.; Wang, Z.; Zhang, M.; Fang, K.; Xu, J.; Liu, Y.; and Ma, S. 2021. WG4Rec: Modeling Textual Content with Word Graph for News Recommendation. In *ACM International Conference on Information and Knowledge Management (CIKM)*, 1651–1660.

Tian, Y.; Yang, Y.; Ren, X.; Wang, P.; Wu, F.; Wang, Q.; and Li, C. 2021. Joint Knowledge Pruning and Recurrent Graph Convolution for News Recommendation. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 51–60.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Conference on Neural Information Processing Systems (NIPS)*, 1–11.

Wang, H.; Zhang, F.; Xie, X.; and Guo, M. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *International Conference on World Wide Web (WWW)*, 1835–1844.

Wang, J.; Chen, Y.; Wang, Z.; and Zhao, W. 2021. Popularity-Enhanced News Recommendation with Multi-View Interest Representation. In *ACM International Conference on Information and Knowledge Management (CIKM)*, 1949–1958.

Wu, C.; Wu, F.; An, M.; Huang, J.; and Huang, Y. 2019a. NPA: Neural News Recommendation with Personalized Attention. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2576–2584.

Wu, C.; Wu, F.; An, M.; Huang, J.; Huang, Y.; and Xie, X. 2019b. Neural News Recommendation with Attentive Multi-View Learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 3863–3869.

Wu, C.; Wu, F.; Ge, S.; Qi, T.; Huang, Y.; and Xie, X. 2019c. Neural News Recommendation with Multi-Head Self-Attention. In *Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6389–6394.

Wu, C.; Wu, F.; Qi, T.; and Huang, Y. 2021a. Two Birds with One Stone: Unified Model Learning for Both Recall and Ranking in News Recommendation. *arXiv preprint arXiv:2104.07404*.

Wu, C.; Wu, F.; Qi, T.; Liu, Q.; Tian, X.; Li, J.; He, W.; Huang, Y.; and Xie, X. 2021b. FeedRec: News Feed Recommendation with Various User Feedbacks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2088–2097.

Wu, F.; et al. 2020. MIND: A Large-scale Dataset for News Recommendation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 3597–3606.

Wu, T. Q. F.; Wu, C.; and Huang, Y. 2021. PP-Rec: News Recommendation with Personalized User Interest and Time-aware News Popularity. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 5457–5467.