**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From the observation that users reading news tend to not click outdated news, we propose the notion of ‘lifetime’ of news, with two hypotheses: (i) news has a shorter lifetime, compared to other types of items such as movies or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 prensent a novel approach for news recommendation, namely, Lifetime-Aware News recommendation 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popular domains of entertainment (e.g., Netflix) and e-commerce (e.g., Amazon), users in news domain rarely click outdated news (Wang et al. 2018; Wu et al. 2020).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:

- **H₁**: 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).
- **H₂**: 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 recommendation 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-
To verify this tendency, we analyzed both datasets of news from Adressa (Gulla et al. 2019c) and movies/dramas from Netflix (Adressa). The period from the initial publish time to the last clicked time, where \( m\% \) of clicks occurred, is a period called the lifetime of news. That is, many recommendations by NRMS (Wu et al. 2019c) still recommend a lot of news after 48 hours from their initial publish time, which is relatively 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 occurred. 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. 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.
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., 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., $d_B$) 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.

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:

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

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; $D$ 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, $d_A^p$, $d_A^q$, $d_B^p$, $d_B^q$, $d_C^p$, and $d_C^q$, that have different periods of lifetimes with each other. Here, $d_A^p$, $d_B^p$, and $d_C^p$ 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: $CPT(d_A^p) = \{ d_B^p \}; CPT(d_B^p) = \{ d_A^p, d_C^p \}; CPT(d_A^q) = \{ d_B^q \}; CPT(d_B^q) = \{ d_A^q, d_C^q \}; CPT(d_B^p) = \{ d_A^p, d_C^p \}; CPT(d_C^q) = \{ d_A^q \}$.

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 \(c_{time1}\), the user \(p\) clicked only \(d^1_{pB}\). So, \(d^1_{pB}\) and \(d^1_{pA}\) are regarded as \(p\)’s positive and negative news, respectively. Then during \(c_{time2}\), since the user \(q\) clicked only \(d^1_{qB}\), \(q\)’s positive and negative news are considered as \(d^1_{qB}\) and \(d^1_{qC}\), respectively. On the other hand, during \(c_{time3}\), since no news were clicked by the users, we are not able to distinguish which one is positive or negative between \(d^1_{pA}\) and \(d^1_{pB}\). During \(c_{time4}\), \(d^2_{qC}\) is regarded to be more preferred than \(d^2_{pB}\) by the user \(p\). Consequently, in our approach, the following positive/negative news for each user are engaged to train models for news recommendations: \(d^1_{pB}/d^1_{pA}\) and \(d^2_{qC}/d^2_{pB}\) for \(p\); \(d^1_{qB}/d^1_{qC}\) 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., \(d^1_{pB}/d^1_{pA}\) and \(d^2_{qC}/d^2_{pB}\) for \(p\); \(d^1_{qB}/d^1_{qC}\) 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 \(d^1_{pB}\) and \(d^1_{qB}\) but not \(d^1_{pA}\). 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 \(d^1_{pA}\) 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:

\[
\text{conf}(u, d_i, d_j) = 1 - \left( \frac{1}{\log(\text{pop}(u, d_i, d_j))} \right),
\]

where \(d_i\) and \(d_j\) indicate a user \(u\)’s positive and negative news, respectively; \(\text{conf}(u, d_i, d_j)\) indicates a confidence in the negativeness of \(d_i\), competing with \(d_j\) by \(u\); \(\text{pop}(u, d_i, d_j)\) 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 \(\text{pop}(u, d_i, d_j)\)). 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:

\[
\mathcal{L}_u = \sum_{d_i \in \mathcal{L}_u} \log \left( \frac{e^{\hat{p}(u, d_i)}}{e^{\hat{p}(u, d_i)} + \sum_{j=1}^{K} e^{\hat{p}(u, d_j)}} \right)
\]

where \(d_i\) and \(d_j\) indicate a user \(u\)’s positive and negative news, respectively; \(\mathcal{L}_u\) 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., \(\bar{u}\) and \(\bar{d}_i\) 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:

\[
\hat{p}(u, d_i, t_{rec}) = \frac{1}{1 + e^{-\alpha |r_{time}(d_i, t_{rec})|}} \cdot \hat{p}(u, d_i) \tag{5}
\]

where \(\hat{p}(u, d_i, t_{rec})\) indicates the adjusted preference for news \(d_i\) at the (current) time of recommendation, \(t_{rec}\); \(\alpha\) indicates a hyperparameter for scaling the degree to which the predicted preference of the news is lowered, according to its length of remaining lifetime; \(|r_{time}(d_i, t_{rec})|\) indicates the length of remaining lifetime of \(d_i\) at \(t_{rec}\), which is determined by \(|t_{time}(d_i) - (t_{rec} - t_{pub}(d_i))|\), 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) 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.

**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?

**Table 1:** Statistics of two real-world datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th># of users</th>
<th># of items</th>
<th># of clicks</th>
<th>Sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adressa</td>
<td>259,709</td>
<td>24,060</td>
<td>6,067,109</td>
<td>99.9%</td>
</tr>
<tr>
<td>MIND</td>
<td>200,000</td>
<td>78,316</td>
<td>4,627,681</td>
<td>99.7%</td>
</tr>
</tbody>
</table>

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).
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 \(p < 0.05\) and \(p < 0.005\) for the paired \(t\)-test with LANCER\(_C/\_N/R\), respectively (Adressa)

<table>
<thead>
<tr>
<th>Metric</th>
<th>LANCER(_C/_N/R)</th>
<th>Gain (%)</th>
<th>LANCER(_C/_N/R)</th>
<th>Gain (%)</th>
<th>LANCER(_C/_N/R)</th>
<th>Gain (%)</th>
<th>LANCER(_C/_N/R)</th>
<th>Gain (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
</tr>
<tr>
<td>Orig(^{(O)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
</tr>
<tr>
<td>MRR</td>
<td>225</td>
<td>255</td>
<td>258</td>
<td>280</td>
<td>24.4</td>
<td>216</td>
<td>223</td>
<td>228</td>
</tr>
<tr>
<td>G@5</td>
<td>208</td>
<td>251</td>
<td>230</td>
<td>280</td>
<td>34.8</td>
<td>196</td>
<td>211</td>
<td>217</td>
</tr>
<tr>
<td>G@10</td>
<td>273</td>
<td>329</td>
<td>333</td>
<td>.363</td>
<td>33.0</td>
<td>283</td>
<td>293</td>
<td>301</td>
</tr>
</tbody>
</table>

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)

<table>
<thead>
<tr>
<th>Metric</th>
<th>LANCER(_C)(^{(1)})</th>
<th>Imp(^{(i)})</th>
<th>Orig(^{(O)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td></td>
</tr>
<tr>
<td>Orig(^{(O)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td></td>
</tr>
<tr>
<td>MRR</td>
<td>688</td>
<td>836</td>
<td>.850</td>
<td>23.6</td>
<td>1.7</td>
<td>587</td>
<td>663</td>
<td>.688</td>
<td>17.2</td>
<td>3.8</td>
<td>732</td>
</tr>
<tr>
<td>G@5</td>
<td>325</td>
<td>520</td>
<td>.526</td>
<td>62.2</td>
<td>1.2</td>
<td>223</td>
<td>351</td>
<td>.352</td>
<td>37.8</td>
<td>0.3</td>
<td>399</td>
</tr>
<tr>
<td>G@10</td>
<td>404</td>
<td>604</td>
<td>.614</td>
<td>52.0</td>
<td>1.7</td>
<td>287</td>
<td>414</td>
<td>.424</td>
<td>47.7</td>
<td>2.4</td>
<td>475</td>
</tr>
</tbody>
</table>

Table 4: Accuracy of the variant from LANCER\(_C\)\(^{(1)}\) that is contrary to Idea2 (Adressa)

<table>
<thead>
<tr>
<th>Metric</th>
<th>LANCER(_C)(^{(1)})</th>
<th>Imp(^{(i)})</th>
<th>Orig(^{(O)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
<th>C(^{(C)})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
<td>(\text{NRMS}^{__})</td>
</tr>
<tr>
<td>Orig(^{(O)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
<td>C(^{(C)})</td>
</tr>
<tr>
<td>MRR</td>
<td>0.615</td>
<td>0.574</td>
<td>0.665</td>
<td>0.621</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G@5</td>
<td>0.242</td>
<td>0.219</td>
<td>0.272</td>
<td>0.244</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G@10</td>
<td>0.312</td>
<td>0.281</td>
<td>0.350</td>
<td>0.316</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 \(\text{NRMS}\), \(\text{LSTUR}\), \(\text{NAML}\), and \(\text{CNE-SUE}\), respectively, where the gain is computed by \((\text{LANCER}_C - \text{Orig})/\text{Orig} \times 100\). 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.

EQ2. To answer EQ2, we also designed the variant LANCER\(_C\)\(^{(1)}\) 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\(_C\)\(^{(1)}\) universally outperform the models equipped with LANCER\(_C\) on Adressa. The accuracies of \(\text{NRMS}\), \(\text{LSTUR}\), \(\text{NAML}\), and \(\text{CNE-SUE}\) are enhanced by up to about 2.7\%, 2.8\%, 13.5\% and 4.6\%, respectively, where the gain is computed by \((\text{LANCER}_C\!-\!\text{LANCER}_C\!^{(1)})/\text{LANCER}_C \times 100\). Also, Table 4 reports the results from the additional variant, LANCER\(_C\)\(^{(1)}\)\(^{-}\), 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 \((\text{LANCER}_C/\_N > \text{LANCER}_C > \text{LANCER}_C\!^{-}\)) regardless of base models. These results indicate the following observations: (i) negative sampling by confidence based on wrong assumption (i.e., LANCER\(_C\)\(^{(1)}\)) can show worse accuracy than ran-
dom sampling without the consideration of confidence for the negative sampling (i.e., $\text{LANCER}_{C/N}$); (ii) our proposed scheme for confidence-based negative sampling with the popularity of news (i.e., $\text{LANCER}_{C/N/R}$) 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 $\text{LANCER}$ of integrating all three key ideas (i.e., $\text{LANCER}_{C/N/R}$) with the variant of $\text{LANCER}_{C/N}$. Here, we set $\alpha$ 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 $\text{LANCER}_{C/N/R}$ consistently outperform the models equipped with $\text{LANCER}_{C/N}$ on Adressa. The models equipped with $\text{LANCER}_{C/N/R}$ improve the accuracy by up to about 16% and 9%, for NAML and CNE-SUE, respectively, compared with the models equipped with $\text{LANCER}_{C/N}$.

with $\text{LANCER}_{C/N}$. Here, the gain is computed by $(\text{LANCER}_{C/N/R} - \text{LANCER}_{C/N})/\text{LANCER}_{C/N} \times 100$. 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., $\text{LANCER}_{C/N}$ and $\text{LANCER}_{C/N/R}$), 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 $\text{LANCER}_{C/N/R}$ than by $\text{LANCER}_{C/N}$. Consequently, our proposed $\text{LANCER}_{C/N/R}$ 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 $\alpha$, 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 $\alpha$ significantly lower the predicted preferences of news with the small length of remaining lifetime.

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

**Conclusion**

In this paper, we explored 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, $\text{LANCER}$, 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 $\text{LANCER}$. 
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


