On the Importance of News Content Representation in Hybrid Neural Session-based Recommender Systems

Gabriel de Souza P. Moreira*
CI&T
Campinas, SP, Brazil
gspmoreira@gmail.com

Dietmar Jannach
University of Klagenfurt
Klagenfurt, Austria
dietmar.jannach@aau.at

Adilson Marques da Cunha
Instituto Tecnológico de Aeronáutica
São José dos Campos, SP, Brazil
cunha@ita.br

ABSTRACT

News recommender systems are designed to surface relevant information for online readers by personalizing their user experiences. A particular problem in that context is that online readers are often anonymous, which means that this personalization can only be based on the last few recorded interactions with the user, a setting named session-based recommendation. Another particularity of the news domain is that constantly fresh articles are published, which should be immediately considered for recommendation. To deal with this item cold-start problem, it is important to consider the actual content of items when recommending. Hybrid approaches are therefore often considered as the method of choice in such settings. In this work, we analyze the importance of considering content information in a hybrid neural news recommender system. We contrast content-aware and content-agnostic techniques and also explore the effects of using different content encodings. Experiments on two public datasets confirm the importance of adopting a hybrid approach. Furthermore, we show that the choice of the content encoding can have an impact on the resulting performance.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**;

KEYWORDS

Recommender Systems; Hybrid Systems; News Recommendation; Session-Based Recommendation; Recurrent Neural Networks

1 INTRODUCTION & BACKGROUND

Many of today's major media and news aggregator websites, including The New York Times [38], The Washington Post [9], Google News [5], and Yahoo! News [39], provide automated reading recommendations for their users. News recommendation, while being one of the earliest application fields of recommenders, is often still considered a challenging problem for a many reasons [16].

Among them, there are two types of cold-start problems. First, there is the permanent *item cold-start problem*. In the news domain, we have to deal with a constant stream of possibly thousands of new articles published each day [38]. At the same time, these articles become outdated very quickly [5]. Second, on many news sites, we have to deal with *user cold-start*, when users are anonymous or not logged-in [7, 22, 25], which means that personalization has to be based on a few observed interactions (e.g., clicks) of the user.

In many application domains of recommenders, collaborative filtering techniques, which only rely on observed preference patterns in a user community, have proven to be highly effective in the past. However, in the particular domain of news recommendation, the use of hybrid techniques, which also consider the actual content of a news item, have often shown to be preferable to deal with item cold-start, see e.g., [2, 8, 22, 23, 25, 26, 37, 39].

Likewise, to deal with user cold-start issues, *session-based recommendation* techniques received more research interest in recent years. In these approaches, the provided recommendations are not based on long-term preference profiles, but solely on adapting recommendations according to the most recent observed interactions of the current user.

Technically, a number of algorithmic approaches can be applied for this problem, from rule-learning techniques, over nearest-neighbor schemes, to more complex sequence learning methods and deep learning approaches. For an overview see [34]. Among the neural methods, Recurrent Neural Networks (RNN) are a natural choice for learning sequential models [12, 21]. Attention mechanisms have also been used for session-based recommendation [27].

The goal of this work is to investigate two aspects of hybrid session-based news recommendation using neural networks. Our first goal is to understand the value of considering content information in a hybrid system. Second, we aim to investigate to what extent the choice of the mechanism for encoding the articles' textual content matters. To that purpose, we have made experiments with various encoding mechanisms, including unsupervised (like *Latent Semantic Analysis* and *doc2vec*) and supervised ones. Our experiments were made using a realistic streaming-based evaluation protocol. The outcomes of our studies, which were based on two public datasets, confirm the usefulness of considering content information. However, the quality and detail of the content representation matters, which means that care of these aspects should be taken in practical settings. Second, we found that the specific document encoding *can* makes a difference in recommendations quality, but sometimes those differences are small. Finally, we found that content-agnostic nearest-neighbor methods,

*Also with Brazilian Aeronautics Institute of Technology.

which are considered highly competitive with RNN-based techniques in other scenarios [14, 28], were falling behind on different performance measures compared to the used neural approach.

2 METHODOLOGY

To conduct our experiments, we have implemented different instantiations of our deep learning *meta-architecture* for news recommendation called *CHAMELEON* [32, 33]. The main component of the architecture is the *Next-Article Recommendation (NAR)* module, which processes various types of input features, including pre-trained *Article Content Embeddings (ACE)* and contextual information about users (e.g., time, location, device) and items (e.g., recent popularity, recency). These inputs are provided for all clicks of a user observed in the current session to generate next-item recommendations based on an RNN (e.g., GRU, LSTM).

The *ACEs* are produced by the *Article Content Representation (ACR)* module. The input to the module is the article's text, represented as a sequence of word embeddings (e.g. using Word2Vec [31]), pre-trained on a large corpus. These embeddings are further processed by *feature extractors*, which can be instantiated as Convolutional Neural Networks (CNN) or RNNs. The *ACR module*'s neural network is trained in a supervised manner for a side task: to predict metadata attributes of an article, such as categories or tags. Figure 1 illustrates how the *Article Content Embeddings* are used within *CHAMELEON*'s processing chain to provide next-article recommendations.

In this work, we first analyzed the importance of considering article content information for recommendations. Second, we experimented with different techniques for textual content representation[1], and investigated how they might affect recommendation quality. The different variants that were tested [2] are listed in Table 1.

For the experiments, *CHAMELEON*'s *NAR* module took the following features as input, described in more detail in [33] [3]: (1) *Article Content Embeddings* (generated by the different techniques presented in Table 1), (2) article metadata (category and author[4]), (3) article context (novelty and recency), (4) user context (city, region, country, device type, operational system, hour of the day, day of the week, referrer).

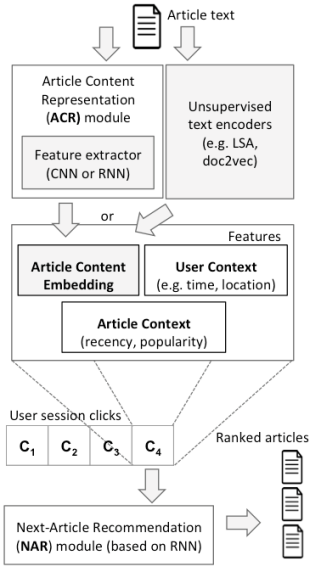

Figure 1: A simplified overview of *CHAMELEON*. The components for which we tested different variants are shaded.

Table 1: Alternative content processing techniques.

Technique	*Input*	*Description*
No-ACE	*None*	In this setting, no content representation is used as input.
Supervised		
CNN	*word2vec*[5]	A 1D-CNN-based model trained to classify the articles' metadata (e.g., category). The architecture combines three CNNs, with window sizes of 3, 4, and 5 to model n-grams. The output of an intermediate layer is used as textual representation. For more details see [32, 33]
GRU	*word2vec*	Similar to the CNN-based version, a GRU layer is trained to classify metadata. The outputs of the GRU layer are max-pooled to generate representations.
Unsupervised		
LSA	Raw text	Traditional *Latent Semantic Analysis (LSA)* [6]. We used a variation based on *TF-IDF* vectors [36] and *Truncated SVD* [11].
*W2V*TF-IDF*	*word2vec*	*TF-IDF* weighted word embeddings [24], a technique to represent a piece of text as the average of its word embeddings weighted by *TF-IDF* [36].
doc2vec	Raw text	Paragraph Vector (a.k.a *doc2vec*) [19] learns fixed-length feature representations from variable-length pieces of texts, which are trained via the distributed memory and distributed bag of words models.

3 EXPERIMENTAL SETUP

We adopt a temporal offline evaluation method as proposed in [32, 33], which simulates a streaming flow of new user interactions (clicks) and articles being published. Since in practical environments it is highly important to quickly react

[1] As there were some very long articles, the text was truncated after the first 12 sentences, and concatenated with the title. *Article Content Embeddings (ACE)* produced by the selected techniques were *L2*-normalized to make the feature scale similar, but also to preserve high similarity scores for embeddings from similar articles.

[2] We also experimented with Sequence *Autoencoders GRU* (adapted from *SA-LSTM* [4]) to extract textual features by reconstructing the sequence of input word embeddings, but this technique did not lead to better results than the other unsupervised methods.

[3] Note that the experiments reported here did not include the *trainable Article ID* feature used in the experiments from [33], which can lead to a slightly improved accuracy, but possibly reduces the differences observed between the content representations.

[4] Article author and user city are available only for the *Adressa* dataset.

[5] Portuguese: A pre-trained Word2Vec [31] *skip-gram* model (300 dimensions) is available at http://nilc.icmc.usp.br/embeddings; and Norwegian: a *skip-gram* model (100 dimensions) is available at http://vectors.nlpl.eu/repository (model #100).

to incoming events [15, 17, 30], the baseline recommender methods are constantly updated over time. *CHAMELEON*'s *NAR* module also supports online learning. The training process of *CHAMELEON* emulates a streaming scenario with mini-batches, in which each user session is used for training only once. Such a scalable approach is different from other techniques, like *GRU4Rec* [12], which require training for some epochs on a larger set of past interactions to reach high accuracy.

3.1 Evaluation Protocol

The evaluation process works as follows:

(1) The recommenders are continuously trained on user sessions ordered by time and grouped by hours. Every five hours, the recommenders are evaluated on sessions from the next hour. With this interval of five hours (not a divisor of 24 hours), we cover different hours of the day for evaluation. After the evaluation of the next hour was done, this hour is also considered for training, until the entire dataset is covered.[6] Note that *CHAMELEON*'s model is only updated after all events of the test hour are processed. This allows us to emulate a realistic production scenario where the model is trained and deployed once an hour to serve recommendations for the next hour;

(2) For each session in the test set, we incrementally reveal one click after the other to the recommender, as done, e.g., in [12, 35];

(3) For each click to be predicted, we sample a random set containing 50 recommendable articles (the ones that received at least one click by any user in the preceding hour) that were *not* viewed by the user in their session (negative samples) plus the true next article (positive sample), as done in [3] and [18]. We then evaluate the algorithms for the task of ranking those 51 items; and

(4) Given these rankings, standard information retrieval (top-n) metrics can be computed.

3.2 Metrics

As relevant quality factors from the news domain [16], we considered accuracy, item coverage, and novelty. To determine the metrics, we took measurements at list length 10. As accuracy metrics, we used the *Hit Rate (HR@n)*, which checks whether or not the true next item appears in the top-n ranked items, and the *Mean Reciprocal Rank (MRR@n)*, a ranking metric that is sensitive to the position of the true next item. Both metrics are common when evaluating session-based recommendation algorithms [12, 15, 28].

Since it is sometimes important that a news recommender not only focuses on a small set of items, we also considered *Item Coverage (COV@n)* as a quality criterion. We computed item coverage as the number of distinct articles that appeared in any top-n list divided by the number of recommendable articles [13]. In our case, the recommendable articles are the ones viewed at least once in the last hour by any user. To measure novelty, we used the *ESI-R@n* metric [33], adapted from [1, 41, 42]. The metric is based on item popularity and returns higher values when long-tail items are among the top-n recommendations.

3.3 Datasets

We use two public datasets from news portals:

(1) *Globo.com* (*G1*) dataset - Globo.com is the most popular media company in Brazil. The dataset[7] was collected at the G1 news portal, which has more than 80 million unique users and publishes over 100,000 new articles per month; and

(2) *SmartMedia Adressa* - This dataset contains approximately 20 million page visits from a Norwegian news portal [10]. In our experiments we used its complete version[8], which includes article text and click events of about 2 million users and 13,000 articles.

Both datasets include the textual content of the news articles, article metadata (such as publishing date, category, and author), and logged user interactions (page views) with contextual information. Since we are focusing on session-based news recommendations and short-term users preferences, it is not necessary to train algorithms for long periods. Therefore, and because articles become outdated very quickly, we have selected all available user sessions from the first 16 days for both datasets for our experiments.

In a pre-processing step, like in [8, 28, 40], we organized the data into sessions using a 30 minute threshold of inactivity as an indicator of a new session. Sessions were then sorted by timestamp of their first click. From each session, we removed repeated clicks on the same article, as we are not focusing on the capability of algorithms to act as reminders as in [20]. Sessions with only one interaction are not suitable for next-click prediction and were discarded. Sessions with more than 20 interactions (stemming from *outlier* users with an unusual behavior or from bots) were truncated.

The characteristics of the resulting pre-processed datasets are shown in Table 2. Coincidentally, the datasets are similar in many statistics, except for the total number of published articles, which is much higher for *G1* than for the *Adressa* dataset.

Table 2: Statistics of the datasets used for the experiments.

	Globo.com (G1)	*Adressa*
Language	Portuguese	Norwegian
Period (days)	16	16
# users	322,897	314,661
# sessions	1,048,594	982,210
# clicks	2,988,181	2,648,999
# articles	46,033	13,820
Avg. session length	2.84	2.70

[6] Our dataset consists of 16 days. We used the first 2 days to learn an initial model for the session-based algorithms and report the averaged measures after this warm-up.

[7] https://www.kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom

[8] http://reclab.idi.ntnu.no/dataset

3.4 Baselines

The baselines used in our experiments are summarized in Table 3. While some baselines appear conceptually simple, recent work has shown that they are often able to outperform very recent neural approaches for session-based recommendation tasks [14, 28, 29]. Unlike neural methods like *GRU4REC*, these methods can be continuously updated over time to take newly published articles into account. A comparison of *GRU4REC* with some of our baselines in a streaming scenario is provided in [15], and specifically in the news domain in [32], which is why we do not include *GRU4REC* and similar methods here.

Table 3: Baseline recommendation algorithms.

Association Rules-based and Neighborhood Methods	
Co-Occurrence (CO)	Recommends articles commonly viewed together with the last read article in previous user sessions [15, 28].
Sequential Rules (SR)	The method also uses association rules of size two. It however considers the sequence of the items within a session and uses a weighting function when two items do not immediately appear after each other [28].
Item-kNN	Returns the most similar items to the last read article using the cosine similarity between their vectors of co-occurrence with other items within sessions. This method has been commonly used as a baseline for neural approaches, e.g., in [12].[9]
Non-personalized Methods	
Recently Popular (RP)	This method recommends the most viewed articles within a defined set of recently observed user interactions on the news portal (e.g., clicks during the last hour). Such a strategy proved to be very effective in the *2017 CLEF NewsREEL Challenge* [30].
Content-Based (CB)	For each article read by the user, this method suggests recommendable articles with similar content to the last clicked article, based on the cosine similarity of their *Article Content Embeddings* (generated by the *CNN* technique described in Table 1).

Replicability. We publish the data and source code used in our experiments online[10], including the code for *CHAMELEON*, which is implemented using *TensorFlow*.

4 EXPERIMENTAL RESULTS

The results for the *G1* and *Adressa* datasets after (hyper-)parameter optimization for all methods are presented[11] in Tables 4 and 5.

Accuracy Results. In general, we can observe that considering content information is in fact highly beneficial in terms of recommendation accuracy. It is also possible to see that the choice of the article representation matters. Surprisingly,

[9]We also made experiments with session-based methods proposed in [28] (e.g. V-SkNN), but they did not lead to results that were better than the *SR* and *CO* methods.

[10]https://github.com/gabrielspmoreira/chameleon_recsys

[11]The highest values for a given metric are highlighted in bold. The best values for the *CHAMELEON* configurations are printed in italics. If the best results are significantly different ($p < 0.001$) from all other algorithms, they are marked with *. We used paired Student's t-tests with Bonferroni correction for significance tests.

Table 4: Results for the G1 dataset.

Recommender	*HR@10*	*MRR@10*	*COV@10*	*ESI-R@10*
CHAMELEON with ACEs generated differently				
No-ACE	0.6281	0.3066	0.6429	6.3169
CNN	0.6585	0.3395	*0.6493*	6.2874
GRU	0.6585	0.3388	0.6484	6.2674
*W2V*TF-IDF*	0.6575	0.3291	0.6500	6.4187
LSA	**0.6686***	**0.3423**	0.6452	6.3833
doc2vec	0.6368	0.3119	0.6431	*6.4345*
Baselines				
SR	0.5911	0.2889	0.2757	5.9743
Item-kNN	0.5707	0.2801	0.3892	6.5898
CO	0.5699	0.2625	0.2496	5.5716
RP	0.4580	0.1994	0.0220	4.4904
CB	0.3703	0.1746	**0.6855***	**8.1683***

Table 5: Results for the Adressa dataset.

Recommender	*HR@10*	*MRR@10*	*COV@10*	*ESI-R@10*
CHAMELEON with ACEs generated differently				
No-ACE	0.6816	0.3252	*0.8185*	5.2453
CNN	0.6860	0.3333	0.8103	5.2924
GRU	0.6856	0.3327	0.8096	5.2861
*W2V*TF-IDF*	0.6913	0.3402	0.7976	5.3273
LSA	**0.6935**	**0.3403**	0.8013	5.3347
doc2vec	0.6898	0.3402	0.7968	*5.3417*
Baselines				
SR	0.6285	0.3020	0.4597	5.4445
Item-kNN	0.6136	0.2769	0.5287	5.4668
CO	0.6178	0.2819	0.4198	5.0785
RP	0.5647	0.2481	0.0542	4.1464
CB	0.3273	0.1197	**0.8807***	**7.6534***

the long-established *LSA* method was the best performing technique to represent the content for both datasets in terms of accuracy, even when compared to more recent techniques using pre-trained word embeddings, such as the *CNN* and *GRU*.

For the *G1* dataset, the *Hit Rates* (*HR*) were improved by around 7% and the *MRR* by almost 12% when using the LSA representation instead of the *No-ACE setting*. For the *Adressa* dataset, the difference between the *No-ACE* settings and the hybrid methods leveraging text are less pronounced. The improvement using *LSA* compared to the *No-ACE setting* was around 2% for *HR* and 5% for *MRR*.

Furthermore, for the *Adressa* dataset, it is possible to observe that all the *unsupervised* methods (*LSA*, *W2V*TF-IDF*, and *doc2vec*) for generating *ACEs* performed better than the *supervised* ones, differently from the *G1* dataset. A possible explanation can be that the *supervised* methods depend more on the *quality* and *depth* of the available article metadata information. While the *G1* dataset uses a fine-grained categorization scheme (461 categories), the categorization of the *Adressa* dataset is much more coarse (41 categories).

Among the baselines, *SR* leads to the best accuracy results, but does not match the performance of the content-agnostic *No-ACE* settings for an RNN. This indicates that the hybrid approach of considering additional contextual information, as done by *CHAMELEON*'s *NAR* module in this condition, is important.

Recommending only based on content information (*CB*), as expected, does not lead to competitive accuracy results, because the popularity of the items is not taken into account

(which *SR* and neighborhood-based methods implicitly do). Recommending only recently popular articles (*RP*) works better than *CB*, but does not match the performance of the other methods.

Coverage and Novelty. In terms of coverage (*COV@10*), the simple *Content-Based (CB)* method leads to the highest value, as it recommends across the entire spectrum based solely on content similarity, without considering the popularity of the items. It is followed by the various *CHAMELEON* instantiations, where it turned out that the specifically chosen content representation is not too important in this respect.

As expected, the *CB* method also frequently recommends long-tail items, which also leads to the highest value in terms of novelty (*ESI-R@10*). The popularity-based method (*RP*), in contrast, leads to the lowest novelty value. From the other methods, the traditional *Item-KNN* method, to some surprise, leads to the best novelty results, even though neighborhood-based methods have a certain popularity bias. Looking at the other configurations, using *unsupervised* methods to represent the text of the articles can help to drive the recommendations a bit away from the popular ones.

5 SUMMARY AND CONCLUSION

The consideration of content information for news recommendation proved to be important in the past, and therefore many hybrid systems were proposed in the literature. In this work, we investigated the relative importance of incorporating content information in both streaming- and session-based recommendation scenarios. Our experiments highlighted the value of content information by showing that it helped to outperform otherwise competitive baselines. Furthermore, the experiments also demonstrated that the choice of the article representation can matter. However, the value of considering additional content information in the process depends on the quality and depth of the available data, especially for *supervised* methods. From a practical perspective, this indicates that quality assurance and curation of the content information can be essential to obtain better results.

REFERENCES

[1] Pablo Castells, Neil J. Hurley, and Saul Vargas. 2015. Novelty and Diversity in Recommender Systems. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer US, 881–918.

[2] Wei Chu and Seung-Taek Park. 2009. Personalized recommendation on dynamic content using predictive bilinear models. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*. 691–700.

[3] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM Conference on Recommender Systems (RecSys '10)*. 39–46.

[4] Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*. 3079–3087.

[5] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web (WWW'07)*. 271–280.

[6] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 6 (1990), 391–407.

[7] Jorge Díez Peláez, David Martínez Rego, Amparo Alonso Betanzos, Óscar Luaces Rodríguez, and Antonio Bahamonde Rionda. 2016. Metrical Representation of Readers and Articles in a Digital Newspaper. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys 2016)*.

[8] Elena Viorica Epure, Benjamin Kille, Jon Espen Ingvaldsen, Rebecca Deneckere, Camille Salinesi, and Sahin Albayrak. 2017. Recommending Personalized News in Short User Sessions. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys'17)*. 121–129.

[9] Ryan Graff. 2015. How the Washington Post used data and natural language processing to get people to read more news. https://knightlab.northwestern.edu/2015/06/03/how-the-washington-posts-clavis-tool-helps-to-make-news-personal/. (June 2015).

[10] Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. 2017. The Adressa dataset for news recommendation. In *Proceedings of the International Conference on Web Intelligence (WI'17)*. 1042–1048.

[11] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* 53, 2 (2011), 217–288.

[12] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of Fourth International Conference on Learning Representations (ICLR'16)*.

[13] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. 2015. What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction* 25, 5 (2015), 427–491.

[14] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys'17)*. 306–310.

[15] Michael Jugovac, Dietmar Jannach, and Mozhgan Karimi. 2018. StreamingRec: A Framework for Benchmarking Stream-based News Recommenders. In *Proceedings of the Twelfth ACM Conference on Recommender Systems (RecSys '18)*. 306–310.

[16] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. 2018. News recommender systems–Survey and roads ahead. *Information Processing & Management* 54, 6 (2018), 1203–1227.

[17] Benjamin Kille, Andreas Lommatzsch, Frank Hopfgartner, Martha Larson, and Torben Brodt. 2017. CLEF 2017 NewsREEL Overview: Offline and Online Evaluation of Stream-based News Recommender Systems. In *Working Notes of CLEF 2017 – Conference and Labs of the Evaluation Forum*.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer* 42, 8 (2009).

[19] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML '14)*. 1188–1196.

[20] Lukas Lerche, Dietmar Jannach, and Malte Ludewig. 2016. On the Value of Reminders within E-Commerce Recommendations. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, (UMAP'16)*.

[21] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. 1419–1428.

[22] Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. 2011. SCENE: a scalable two-stage personalized news recommendation system. In *Proceedings of the 34th International Conference on Research and Development in Information Retrieval (SIGIR'11)*. 125–134.

[23] Lei Li, Li Zheng, Fan Yang, and Tao Li. 2014. Modeling and broadening temporal user interest in personalized news recommendation. *Expert Systems with Applications* 41, 7 (2014), 3168–3177.

[24] Joseph Lilleberg, Yun Zhu, and Yanqing Zhang. 2015. Support vector machines and word2vec for text classification with semantic features. In *14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC '15)*. 136–140.

[25] Chen Lin, Runquan Xie, Xinjun Guan, Lei Li, and Tao Li. 2014. Personalized news recommendation via implicit social experts. *Information Sciences* 254 (2014), 1–18.

[26] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th International Conference on Intelligent User Interfaces (IUI '10)*. 31–40.

[27] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, (KDD '18)*. 1831–1839.

[28] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of Session-based Recommendation Algorithms. *User-Modeling and User-Adapted Interaction* 28, 4–5 (2018), 331–390.

[29] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019. Performance Comparison of Neural and Non-Neural Approaches to Session-based Recommendation. In *Proceedings of the 2019 ACM Conference on Recommender Systems (RecSys 2019)*.

[30] Cornelius A Ludmann. 2017. Recommending News Articles in the CLEF News Recommendation Evaluation Lab with the Data Stream Management System Odysseus.. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF'17)*.

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems (NIPS '13)*. 3111–3119.

[32] Gabriel de Souza Pereira Moreira, Felipe Ferreira, and Adilson Marques da Cunha. 2018. News Session-Based Recommendations using Deep Neural Networks. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems (DLRS) at ACM RecSys'18*. 15–23.

[33] Gabriel de Souza Pereira Moreira, Dietmar Jannach, and Adilson Marques da Cunha. 2019. Contextual Hybrid Session-based News Recommendation with Recurrent Neural Networks. *arXiv preprint arXiv:1904.10367* (2019).

[34] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 66.

[35] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys'17)*. 130–137.

[36] Juan Ramos. 2003. Using TF-IDF to determine word relevance in document queries. In *Technical Report, Department of Computer Science, Rutgers University*.

[37] Junyang Rao, Aixia Jia, Yansong Feng, and Dongyan Zhao. 2013. Personalized news recommendation using ontologies harvested from the web. In *International Conference on Web-Age Information Management*. 781–787.

[38] A. Spangher. 2015. Building the Next New York Times Recommendation Engine. https://open.blogs.nytimes.com/2015/08/11/building-the-next-new-york-times-recommendation-engine/. (Aug 2015).

[39] Michele Trevisiol, Luca Maria Aiello, Rossano Schifanella, and Alejandro Jaimes. 2014. Cold-start news recommendation with domain-dependent browse graph. In *Proceedings of the 8th ACM Conference on Recommender Systems (RecSys'14)*. 81–88.

[40] Bartlomiej Twardowski. 2016. Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys'16)*. 273–276.

[41] Saúl Vargas. 2015. *Novelty and Diversity Evaluation and Enhancement in Recommender Systems*. PhD thesis. Universidad Autónoma de Madrid.

[42] Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM Conference on Recommender Systems (RecSys'11)*. 109–116.