News Session-Based Recommendations using Recurrent Neural Networks

Gabriel de Souza Pereira Moreira
CI&T | Brazilian Aeronautics Institute of Technology - ITA
Sao Jose dos Campos, SP, Brazil
gabrielpm@ciandt.com

Felipe Ferreira
Globo.com
Rio de Janeiro, RJ, Brazil
felipe.ferreira@corp.globo.com

Adilson Marques da Cunha
Brazilian Aeronautics Institute of Technology - ITA
Sao Jose dos Campos, SP, Brazil
cunha@ita.br

ABSTRACT

News recommender systems are aimed to personalize users experiences and help them discover relevant articles from a large and dynamic search space. Therefore, news domain is a challenging scenario for recommendations, due to its sparse user profiling, fast growing number of items, accelerated item's value decay, and users preferences dynamic shift.

Some early promising results have been achieved in the usage of Deep Learning for Recommender Systems, specially for item's feature extraction and for session modeling with Recurrent Neural Networks.

In this paper, its presented a Deep Learning architecture for News Session-Based Recommender Systems, composed of two modules. The first module is responsible to learn news articles representations, based on their text and metadata, whilst the second module provides Session-Based recommendations using Recurrent Neural Networks.

In such extreme cold-start scenario, user context and session information are explicitly modeled. Users' behavior and item features are both merged in an hybrid session-based recommendation approach. The recommendation task addressed in this work is next-item prediction for user sessions: "what is the next most likely article a user might read in a session?"

A time-aware evaluation method is also proposed as a complementary contribution, for a more realistic offline evaluation of such task, considering dynamic factors that affect global readership interests like popularity, recency, and seasonality.

KEYWORDS

Recommender Systems; Deep Learning; News Recommendation; Session-Based Recommendation; Context-Based Recommendation

1 INTRODUCTION

Recommender Systems (RS) have been increasingly popular in assisting users with their choices, thus enhancing their engagement and overall satisfaction with online services [31]. They are an important part of information and e-commerce systems, enabling users to filter through large information and product spaces.

Recommender systems have been researched and applied in online services from different domains, like music [10] [54] [58] (e.g., Spotify, Pandora, Last.fm), videos (e.g. YouTube [15]), people [2] (e.g., Facebook), jobs [4] (e.g., LinkedIn [32], Xing [41]), and research papers [56] [5] (e.g., Docear [6]), among others.

1.1 Deep Learning on Recommender Systems

Deep Learning (DL) [26] [27] [8] [7] is a hot area in machine learning communities. The uptake of deep learning by RS community was relatively slow, as the topic became popular only in 2016, with the first Deep Learning for Recommender Systems workshop at the ACM RecSys 2016 [24].

Early pioneer work applying used neural networks to RS was done in [48], where a two-layer Restricted Boltzmann Machine (RBM) slightly outperformed Matrix Factorization.

After a winter on RS research using neural networks, Deep Collaborative Filtering was addressed by [57] and [60] using denoising auto-encoders [55]. Deep neural networks have recently been used to learn item features from unstructured data, like text [3], music [54] [58], and images [40] [21].

Recurrent Neural Networks (RNN) possess several properties that make them attractive for sequence modeling of user sessions. In particular, they are capable of incorporating input from past consumption events, allowing to derive a wide range of sequence-to-sequence mappings [17]. After the seminal work of [23], a research line has emerged on the usage of RNNs on session-based [25] [22] [59] [39] [49] and session-aware [17] [43] [47] recommendations.

1.2 News Recommender Systems

Popular news portals, such as Google News [14], Yahoo! News [52], The New York Times [51], Washington Post [44] [9], among others, have gained increasing attention from a massive amount of online news readers.

Online news recommendations have been addressed by researchers in the last years, either using Content-Based Filtering [35] [11] [46] [29] [42], Collaborative Filtering [14] [16], and Hybrid approaches [13] [38] [35] [45] [37] [36] [53] [19].

The news domain poses some challenges for Recommender Systems:

- **Sparse user profiling** – the majority of readers are anonymous, and they actually read only few stories from the entire

DLRS 2018, Vancouver, Canada

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

repository. This results in extreme levels of sparsity in the user-item matrix, and users usually have tracked very few information about past behaviour, if any [35] [37] [16];

- **Fast growing number of items** – hundreds of new stories are added daily in news portal (e.g., over 300 in New York Times [51]). This intensifies the cold-start problem, as for fresh items you cannot count on lots of interactions before starting to recommend them [16]. For news aggregators, scalability problems may arise, as a high volume of news articles overload the web within limited time span [42];
- **Accelerated item's value decay** – information value decays over time. This is specially true in the news domain, as most users are interested in fresh information. Thus, each item is expected to have a short shelf life [14]; and
- **Users preferences shift** - users interests on news are not as stable as in the entertainment domain. Some user interests shift over time, while other long-term interests remain stable [16]. User's current interest in a session may be affected by his context (e.g., location, access time) [16] or by global context (e.g., breaking news or important events) [19].

2 A DEEP LEARNING ARCHITECTURE FOR NEWS SESSION-BASED RECOMMENDATIONS

The main contribution of this work is to propose a Deep Learning Architecture for News Session-Based Recommendation composed of two complementary modules, with independent life cycles for training and inference: the Article Content Representation (ACR) and the Next-Article Recommendation (NAR) modules.

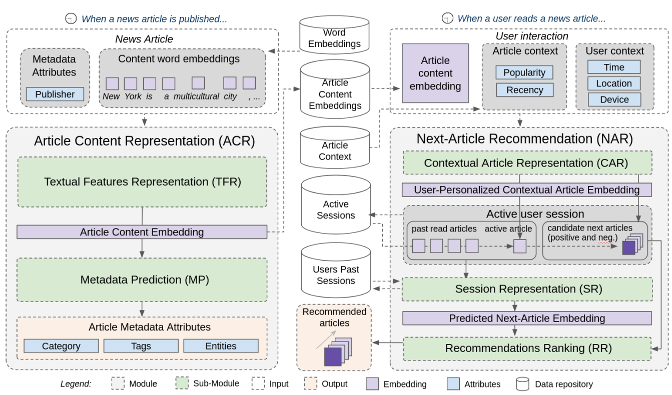

Figure 1: CHAMELEON - The Deep Learning Meta-Architecture for News Recommender Systems

2.1 Article Content Representation (ACR) module

The ACR module is responsible to extract features from news articles text and metadata and learn a distributed representation (embeddings) for each news article context.

The inputs for the ACR module are (1) article metadata attributes (e.g., publisher) and (2) article textual content, represented as a sequence of word embeddings.

Pre-training word embeddings in a larger text corpus of the target language (e.g., Wikipedia) is a common practice in Deep NLP, by using methods like Word2Vec and GloVe.

Textual features are extracted by means of 1D CNNs, which usually present good performance in Deep NLP architectures, like in [34] and [12].

Article's textual features and metadata inputs are combined by using a sequence of Fully Connected (FC) layers to produce the Article Content Embedding.

For scalability reasons, the Article Content Embeddings are not directly trained for recommendation task, but for a side task of news metadata classification. For this work, the Article Content Embeddings were trained to predict the category (editorial section) of the articles.

Softmax function was used to normalize the output layer as a probability distribution that sums up to one, as follows:

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}. \tag{1}$$

Cross-entropy log loss is used for optimization, as follows

$$l(\theta) = -\frac{1}{N}(\sum_{i=1}^{N} y_i \cdot log(\hat{y}_i)) + \lambda\|\theta\|, \tag{2}$$

where y is a vector with the one-hot encoded label for each instance, $\hat{y}$ is the vector with the output probabilities for each class, previously normalized by softmax, θ, representing model parameters to be learned, and λ to control the importance of the regularization term, to avoid overfitting.

After the training of Article Content Embeddings, they are stored in a repository, for further usage by NAR module.

2.2 Next-Article Recommendation (NAR) module

The Next-Article Recommendation (NAR) module is responsible to provide news articles recommendations for active sessions.

Due to the high sparsity of users and their constant interests shift, this work leverages only session-based contextual information, ignoring possible users' past sessions.

The inputs for the NAR module are: (1) the pre-trained *Article Content Embedding* of the last interacted article; (2) the contextual properties of the article (popularity and recency); and (3) the user context (e.g. time, location, and device). These inputs are combined by Fully Connected layers to produce a *User-Personalized Contextual Article Embedding*, whose representations might differ for the same article, depending on the user context and on the current article context (popularity and recency).

The NAR model uses a RNN layer to model the sequence of articles read by users in their sessions, represented by its *User-Personalized Contextual Article Embeddings*. For each article of the sequence, the RNN outputs a *Predicted Next-Article Embedding* – the expected representation of a news content the user would like to read next in the active session.

Most deep learning architectures proposed for RS, like X< Y, Z, the neural network outputs a vector whose dimension is the number of available items. Such approach may work for domains were the items number is more stable, like movies and books. Although, in the dynamic scenario of news recommendations, where thousands of news stories are added and removed daily, such approach would require full retrain the network, as often as new articles are published.

For this reason, the NAR module aims to maximize the similarity between the *Predicted Next-Article Embedding*, and the *User-Personalized Contextual Article Embedding* corresponding to the next article actually read by the user in his session (ground truth), whilst minimizing its similarity with negative samples (articles not read by the user during its session).

With this strategy, a newly published article might be immediately recommended, as soon as its Article Content Embeddings is trained and added to a repository. The inspiration for this approach came from the DSSM [28] and derived works from the RS, like the MV-DNN [18], the TDSSM [50], and the RA-DSSM [33], which uses a training loss based on embeddings similarity.

The *Predicted Next-Article Embedding* and the *User-Personalized Contextual Article Embedding*, further referred as s and $item$, respectively, are vectors with the same arbitrary dimension. In Equation 3, it is defined the relevance function r of an $item$ to a user session s as their cosine similarity. The Cosine similarity of vectors a and b consists of the cosine of the angle between the vectors, as shown in Equation 4.

$$R(s, item) = cos(s, item) \tag{3}$$

$$cos(\theta) = \frac{a \cdot b}{\|a\|\|b\|} \tag{4}$$

Ranking-based loss functions are usually suitable for Top-N recommendations. The objective of the NAR module is to produce a ranked list of the next likely article ($item \in D$) the user will read in the session (a.k.a next-click prediction). Thus, the model should learn to maximize the similarity between the *Predicted next-article Embedding* (s) and the *User-Personalized Contextual Article Embedding* of the next article read by the user ($item^+$), whilst minimizing the pairwise similarity between s and other articles not read by the user ($item^- \in D^-$), naively assuming they are not relevant to the user.

As D set may be a very large in news domain, it is approximated as a set D', by the union of the unit set of the clicked item $\{item^+\}$ with a random negative sample of items not read by the user D^-.

The posterior probability of a read article given a user session can be computed by using a softmax function over the relevance score, proposed for the DSSM [28], as shown in Equation 5,

$$P(item+ \mid s) = \frac{exp(\gamma R(s, item+)}{\sum_{\forall item \in D'} exp(\gamma R(s, item)} \tag{5}$$

where γ is a smoothing factor in the softmax function, which may be empirically set on a held-out data set.

The NAR module neural network parameters are estimated to maximize the likelihood of the next-clicked articles given user sessions. Thus, the loss function to be minimized was also introduced for the DSSM [28], as shown in Equation 6,

$$l(\theta) = -log \prod_{(s, item^+)} P(item^+ \mid s), \tag{6}$$

where θ represent the model parameters to be learned. Since $L(\theta)$ is differentiable w.r.t. to θ, the NAR module is trained using back-propagation on gradient-based numerical optimization algorithms.

3 EXPERIMENTS

The proposed Deep Learning Architecture for News Session-Based Recommendations, composed of ACR and NAR modules, were implemented using TensorFlow [1] [1].

For these experiments, it was used a private dataset provided by Globo.com [2], the most popular news portals in Brazil with more than 80M unique users and 100k new contents per month. The dataset sample contained user interactions during Oct. 1-16, 2017, including more than 3 million clicks, distributed in 1.2 million sessions from 330,000 users, which read more than 50,000 different news articles during that period.

3.1 ACR module training and evaluation

The ACR module was used to learn the *Article Content Embeddings* for news articles. The model was trained to classify the article category (editorial subsection in the news portal) based on its textual content and metadata.

Articles text was represented by sequences of pre-trained word embeddings for Portuguese language [3]. Textual representation was learned by three 1D CNN layers, with window sizes of 3, 4, and 5, which are concatenated with other article's metadata (publisher) by means of a Fully Connected layer.

Training and evaluation were performed using the same dataset, as the objective for this network was not generalization, but to learn representations for articles content and metadata (*Article Content Embeddings*).

Figure 2 presents a visualization, produced using t-SNE, with sampled article embeddings for the 15 categories with most articles. It can be observed that articles are clustered around their categories, which is expected, since the embeddings were trained to classify the articles categories.

After the training, *Article Content Embeddings* were persisted in a repository, for further usage by NAR module.

3.2 NAR module training and evaluation

News readership is very dynamic, as global interests may suddenly shift due to breaking events (e.g., natural disasters, or a real family member birth) or may follow some **seasonality** (e.g., soccer during the World Cup, politics during a presidential election, etc.) [13] [20]. The popularity of a news article often usually very fast, in function of hours.

For this reason, it was devised for NAR module a method for temporal offline evaluation, which mimics a real-world scenario of continuously training the model with streaming user clicks and deploying a new trained model each hour, to provide recommendations for real users, as follows.

[1]For reviewers: the solution code is being refactored and will be open-sourced by the author in a few weeks on GitHub

[2]http://g1.globo.com/

[3]It was used a 300-dims Word2Vec *skip-gram* model, available in http://nilc.icmc.usp.br/embeddings

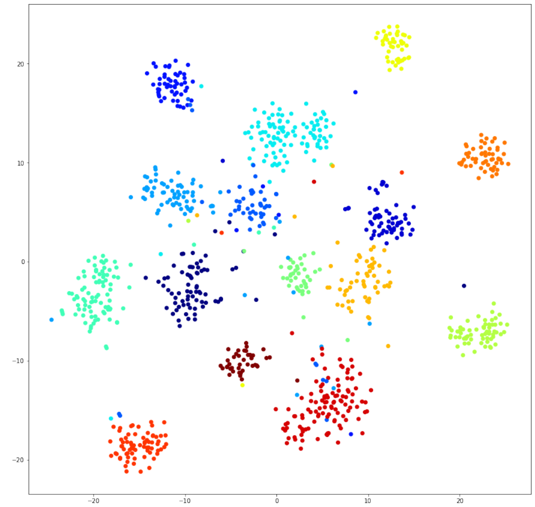

Figure 2: t-SNE visualization of sampled *Article Content Embeddings*, for the top 15 categories

(1) Trains the NAR module with the sessions within the active hour;
(2) Evaluate the NAR module with sessions within the next hour, for the task of next-click prediction.

This method is also scalable because, as each session is used only once for NAR module training (online learning), there is no need to process past sessions again (full retrain).

To keep the consistency of temporal contextual information (popularity and recency) of articles during training and evaluation, the following method was created:

(1) Keeps a global buffer with the last N clicks (article reads), considering all users.
(2) Computes articles recent popularity by counting their clicks in the buffer.
(3) For each article read by a user, uses as input for Deep News the articles dynamic context: recent popularity and recency (number of hours since article was published).

The NAR module requires a number of negative samples to perform training and evaluation. The strategy adopted was to consider as negative samples any article, not read in the session, which was read in other sessions within the training/evaluation mini-batch. When there are not enough negative samples within the batch, they are sampled from the last-clicks buffer. The purpose of this approach is to train the network to differentiate the next clicked item (positive item) from other strong negative candidates (articles recently clicked by other users).

For training, 7 negative samples were used for each session, and for evaluation, 50 negative samples. Evaluation metrics were a top-N accuracy metric and a ranking metric: Recall@5 and NDCG@5 [30], respectively.

3.3 Benchmark methods

For this experiment, it was used a set of benchmark recommendation methods, commonly used in news portals. These methods have the advantage of being updated over time to match current users global interests, and usually have competitive accuracy.

- **Co-occurrent** - For each article read by a user, recommends other articles commonly read together in other user sessions.
- **Recently Popular** - Recommends most popular articles from a global buffer with the last N users clicks.
- **Content-Based** - For each article read by the user, recommends similar articles, based on the cosine similarity of their *Article Content Embeddings*.

To keep results comparable, recommendations provided by benchmarks were filtered to the same sets of 50 negative samples by session, used to evaluate the DeepNews model.

3.4 Results

Two experiments were performed in this study, involving different time periods and evaluation frequency:

(1) Continuous training and evaluation (each five hours) during 15 days.
(2) Continuous training and evaluation (each hour) of the last day (Oct. 16, 2017).

The Recall@5 and NDCG@5 were accumulated and reported by hour.

3.4.1 Experiment 1. For this experiment, DeepNews model was trained sequentially on user sessions during 15 days. Required data for benchmarks methods (e.g. articles recent popularity, co-ocurrences of articles within sessions) was also updated online at each click, to emulate a live environment.

After training on sessions within five hours, sessions of the next hour were used for evaluation.

Figure 3 presents the evolution of Recall@5 over time, for the sampled evaluation hours. It can be seen that DeepNews have constantly a higher accuracy than the benchmark methods.

The distribution of Recall@5 by hour can be viewed in Figure 4. The median Recall@5 for DeepNews was X, while the best benchmark got a median of Z.

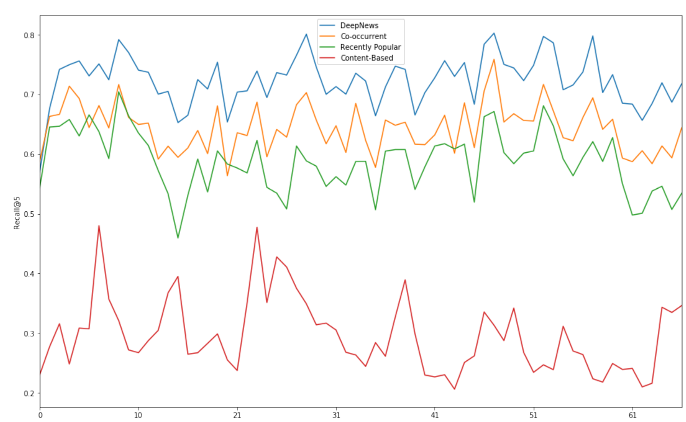

Figure 3: Recall@5 by hours (sampled for evaluation), during 15 days

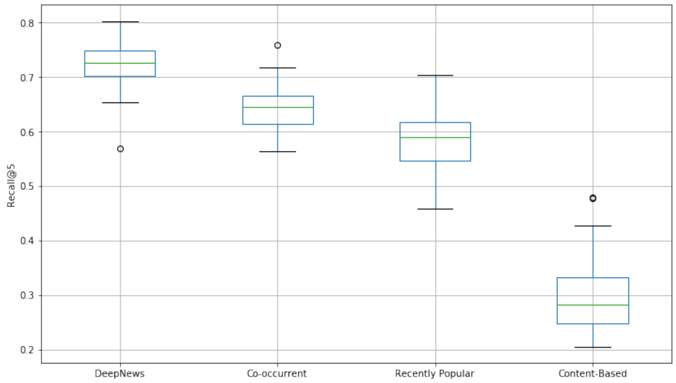

Figure 4: Distribution of Recall@5 by hour (sampled for evaluation), for 15 days

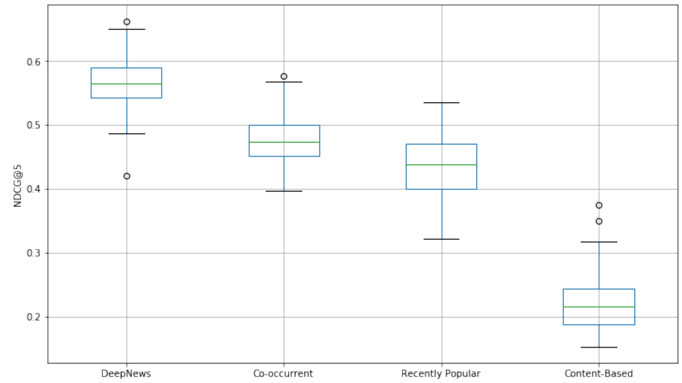

Figure 6: Distribution of NDCG@5 by hours (sampled for evaluation), for 15 days

The phenomenon is similar with NDCG@5. Figures 5 and 6 show that the ranking of recommendations provided by DeepNews was constantly better than benchmarks methods.

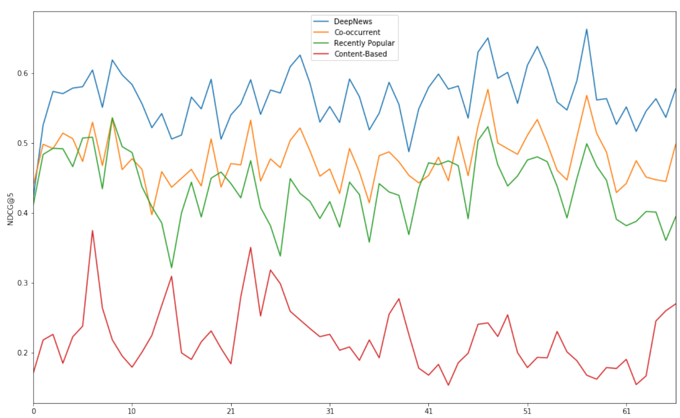

Figure 5: NDCG@5 by hours (sampled for evaluation), during 15 days

3.4.2 Experiment 2. In this experiment, recommendation methods were evaluated in more detail, for a period of 24 hours (Oct. 16, 2017). Thus, after training incrementally on sessions within each hour, sessions of the next hour were used for evaluation.

For this experiment, it was included a Collaborative Filtering method (SVD Matrix Factorization). It is not a session-based approach, but a very popular algorithm in Recommender Systems.

The user-item matrix was initially filled with users clicks from the previous 15 days. Then, a Matrix Factorization model was trained incrementally and evaluated using the temporal offline evaluation method.

Figures 7 and 8 presents the evolution of Recall@5 and NDCG@5 by hour, within a period of 24 hours. Once again, it can be seen that DeepNews accuracy and ranking quality kept higher than the benchmarks methods.

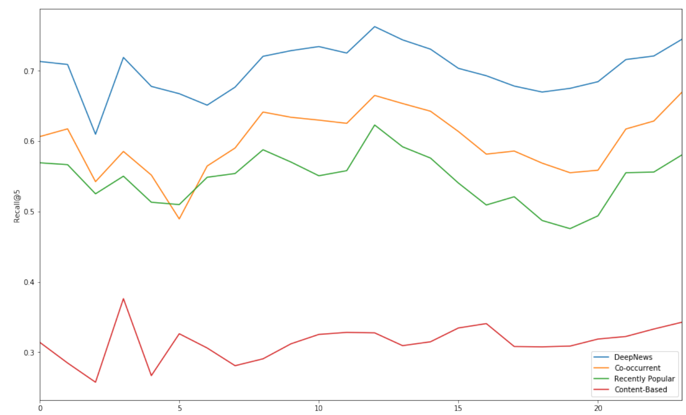

Figure 7: Recall@5 by hour for Oct. 16, 2017

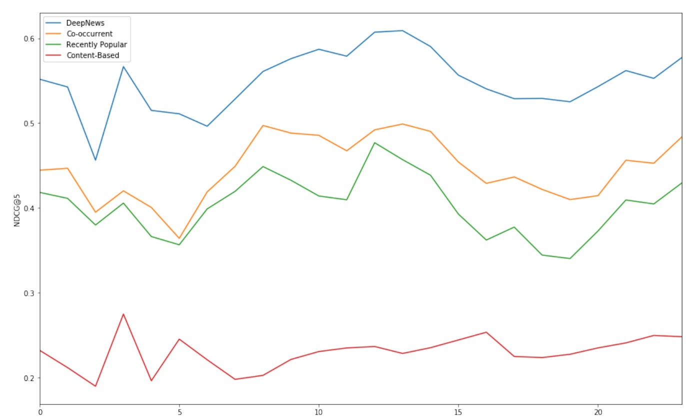

Figure 8: NDCG@5 by hour for Oct. 16, 2017

3.5 Comparisons with Related Work

One of the inspirations for this study came from the Multi-View Deep Neural Network (MV-DNN) [18], which adapted Deep Structured Semantic Model (DSSM) [28] for the recommendation task.

The MV-DNN maps users and items to a latent space, where the cosine similarity between users and their preferred items is maximized. That approach makes it possible to keep the neural network architecture static, rather than adding new units into the output layer for each new item (e.g., published article), as required in [23] (softmax loss function).

Other main inspiration was the GRU4Rec [23], the seminal work on the usage of Recurrent Neural Networks (RNN) on session-based recommendations, and subsequent work [25] [22].

The aforementioned works on the RNN session-based/aware architectures mostly trust on a fixed set of item IDs to provide recommendations. We

The aforementioned works on the RNN session-based/aware architectures mostly trust on a fixed set of item IDs to provide recommendations. The DeepNews architecture uses articles content embeddings to represent items and also user context to represent users, dealing smoothly with the incoming stream of new articles and users interactions.

The aforementioned works on the RNN session-based/aware architectures mostly trust on static user IDs and item IDs to provide recommendations. The DeepNews uses articles content embeddings to represent items and also user context to represent users, dealing smoothly with the incoming stream of new articles and users interactions.

Other main inspirations came from the Multi-View Deep Neural Network (MV-DNN) [18], which adapted Deep Structured Semantic Model (DSSM) [28] for the recommendation task.

The MV-DNN maps users and items to a latent space, where the cosine similarity between users and their preferred items is maximized. That approach makes it possible to keep the neural network architecture static, rather than adding new units into the output layer for each new item (e.g., published article), as required in [23] (softmax loss function).

To the best of the knowledge from this research so far, the only works presenting a deep learning architecture for news recommendation were [50] and [33].

The MV-DNN was adapted for News recommendation by [50] Temporal DSSM (TDSSM) and [33] Recurrent Attention DSSM (RA-DSSM). Differently from CHAMELEON, TDSSM [50] did not model user sessions explicitly, and items and users representations are not directly learned from news content and users behaviours.

The RA-DSSM [33] ignores past user sessions information, whilst CHAMELEON provides session-aware news recommendations. Furthermore, whilst RA-DSSM represents articles content by using Doc2Vec embeddings (unsupervised training), CHAMELEON trains news content embeddings to predict news metadata by using multi-task supervised learning. Finally, the RA-DSSM does not use any contextual information about the user and articles, which may limit its accuracy in a extreme cold-start scenario like news RS.

4 CONCLUSIONS

...

ACKNOWLEDGMENTS

The authors would like to thank Globo.com for providing context on its challenges on large-scale news recommender systems and for sharing a dataset to make those experiments possible.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[2] Hernan Badenes, Mateo N Bengualid, Jilin Chen, Liang Gou, Eben Haber, Jalal Mahmud, Jeffrey W Nichols, Aditya Pal, Jerald Schoudt, Barton A Smith, et al. 2014. System U: automatically deriving personality traits from social media for people recommendation. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 373–374.

[3] Trapit Bansal, David Belanger, and Andrew McCallum. 2016. Ask the gru: Multi-task learning for deep text recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 107–114.

[4] Mathieu Bastian, Matthew Hayes, William Vaughan, Sam Shah, Peter Skomoroch, Hyungjin Kim, Sal Uryasev, and Christopher Lloyd. 2014. LinkedIn skills: large-scale topic extraction and inference. In *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 1–8.

[5] Joeran Beel, Stefan Langer, Marcel Genzmehr, Bela Gipp, Corinna Breitinger, and Andreas Nürnberger. 2013. Research paper recommender system evaluation: A quantitative literature survey. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*. ACM, 15–22.

[6] Joeran Beel, Stefan Langer, Marcel Genzmehr, and Andreas Nürnberger. 2013. Introducing Docear's research paper recommender system. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 459–460.

[7] Yoshua Bengio et al. 2009. Learning deep architectures for AI. *Foundations and trends® in Machine Learning* 2, 1 (2009), 1–127.

[8] Yoshua Bengio, Yann LeCun, et al. 2007. Scaling learning algorithms towards AI. *Large-scale kernel machines* 34, 5 (2007), 1–41.

[9] R. Bilton. 2016. The Washington Post tests personalized "pop-up" newsletters to promote its big stories. http://www.niemanlab.org/2016/05/the-washington-post-tests-personalized-pop-up-newsletters-to-promote-its-big-stories/. (May 2016).

[10] Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, and Xiaofei He. 2010. Music recommendation by unified hypergraph: combining social media information and music content. In *Proceedings of the international conference on Multimedia*. ACM, 391–400.

[11] Michel Capelle, Flavius Frasincar, Marnix Moerland, and Frederik Hogenboom. 2012. Semantics-based news recommendation. In *Proceedings of the 2nd international conference on web intelligence, mining and semantics*. ACM, 27.

[12] Rose Catherine and William Cohen. 2017. TransNets: Learning to Transform for Recommendation. *arXiv preprint arXiv:1704.02298* (2017).

[13] 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*. ACM, 691–700.

[14] 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*. ACM, 271–280.

[15] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 293–296.

[16] 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 *10th ACM Conference on Recommender Systems (RecSys 2016)*. ACM.

[17] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential User-based Recurrent Neural Network Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 152–160.

[18] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 278–288.

[19] 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*. ACM, 121–129.

[20] Jon Atle Gulla, Cristina Marco, Arne Dag Fidjestøl, Jon Espen Ingvaldsen, and Özlem Özgöbek. 2016. The Intricacies of Time in News Recommendation.. In *UMAP (Extended Proceedings)*.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[22] Balázs Hidasi and Alexandros Karatzoglou. 2017. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. *arXiv preprint arXiv:1706.03847* (2017).

[23] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of Forth International Conference on Learning Representations*.

[24] Balázs Hidasi, Alexandros Karatzoglou, Oren Sar-Shalom, Sander Dieleman, Bracha Shapira, and Domonkos Tikk. 2017. DLRS 2017-Second Workshop on Deep Learning for Recommender Systems. In *Proceedings of the 2st Workshop on Deep Learning for Recommender Systems. ACM*, Vol. 29. 34.

[25] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. 2016. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 241–248.

[26] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18, 7 (2006), 1527–1554.

[27] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[28] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2333–2338.

[29] Ilija Ilievski and Sujoy Roy. 2013. Personalized news recommendation based on implicit feedback. In *Proceedings of the 2013 international news recommender systems workshop and challenge*. ACM, 10–15.

[30] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[31] Gawesh Jawaheer, Peter Weller, and Patty Kostkova. 2014. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 4, 2 (2014), 8.

[32] Krishnaram Kenthapadi, Benjamin Le, and Ganesh Venkataraman. 2017. Personalized Job Recommendation System at LinkedIn: Practical Challenges and Lessons Learned. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 346–347.

[33] Vaibhav Kumar, Dhruv Khattar, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Neural Architecture for News Recommendation. In *Working Notes of the 8th International Conference of the CLEF Initiative, Dublin, Ireland. CEUR Workshop Proceedings*.

[34] Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Quote Recommendation in Dialogue using Deep Neural Network. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 957–960.

[35] 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 ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 125–134.

[36] 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.

[37] 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.

[38] 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*. ACM, 31–40.

[39] Qiang Liu, Shu Wu, Diyi Wang, Zhaokang Li, and Liang Wang. 2016. Context-aware sequential recommendation. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 1053–1058.

[40] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 43–52.

[41] Sonu K Mishra and Manoj Reddy. 2016. A bottom-up approach to job recommendation system. In *Proceedings of the Recommender Systems Challenge*. ACM, 4.

[42] Itishree Mohallick and Özlem Özgöbek. 2017. Exploring privacy concerns in news recommender systems. In *Proceedings of the International Conference on Web Intelligence*. ACM, 1054–1061.

[43] 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*.

[44] Graff. R. 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).

[45] 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*. Springer, 781–787.

[46] Hongda Ren and Wei Feng. 2013. Concert: A concept-centric web news recommendation system. In *International Conference on Web-Age Information Management*. Springer, 796–798.

[47] Massimiliano Ruocco, Ole Steinar Lillestøl Skrede, and Helge Langseth. 2017. Inter-Session Modeling for Session-Based Recommendation. In *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*. ACM, 24–31.

[48] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*. ACM, 791–798.

[49] Elena Smirnova and Flavian Vasile. 2017. Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks. In *Proceedings of the 11th ACM Conference on Recommender Systems*.

[50] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 909–912.

[51] 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).

[52] Michele Trevisiol, Luca Maria Aiello, Rossano Schifanella, and Alejandro Jaimes. 2014. Cold-start news recommendation with domain-dependent browse graph. In *Proceedings of the ACM Recommender System conference, RecSys*, Vol. 14.

[53] 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*. ACM, 81–88.

[54] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in neural information processing systems*. 2643–2651.

[55] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*. ACM, 1096–1103.

[56] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 448–456.

[57] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1235–1244.

[58] Xinxi Wang and Ye Wang. 2014. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 627–636.

[59] Caihua Wu, Junwei Wang, Juntao Liu, and Wenyu Liu. 2016. Recurrent neural network based recommendation for time heterogeneous feedback. *Knowledge-Based Systems* 109 (2016), 90–103.

[60] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 153–162.