Postprint of: Chłodziński G., Woźniak K., Deep learning for recommending subscription-limited documents, 13th International Conference on Human System Interaction (HSI), 2020, pp. 29-33, DOI: 10.1109/HSI49210.2020.9142663



# Deep learning for recommending subscription-limited documents

Grzegorz Chłodziński
*Department of Algorithms and System Modeling*
*Gdańsk University of Technology*
Gdańsk, Poland
grzegorz.chlodzinski@pg.edu.pl

Karol Woźniak
*IHS Markit*
Gdańsk, Poland
karol.wozniak@ihsmarkit.com

***Abstract*—Documents recommendation for a commercial, subscription-based online platform is important due to the difficulty in navigation through a large volume and diversity of content available to clients. However, this is also a challenging task due to the number of new documents added every day and decreasing relevance of older contents. To solve this problem, we propose deep neural network architecture that combines autoencoder with multilayer perceptron in a hybrid recommender system. We train our model using real-world historical data from commercial platform using interactions to capture user similarity and categorical document features to predict the probability of a user-document interaction. Our experimental results demonstrate the effectiveness of the proposed architecture. We plan to release our model in a commercial online platform to support a personalized user experience.**

***Keywords*—Recommender Systems; News Recommendation; Deep Learning; Autoencoders***

## I. INTRODUCTION

The growth of textual content delivered through commercial, subscription-based online services poses a challenge both for content providers and content consumers. Content providers must be able to distribute and deliver the increased variety and volume of content to different content consumers. Content consumers must be given efficient means of finding relevant content that is part of their paid subscription. This task is challenging as hundreds of new documents are published every day.

This is one of the challenges for IHS Markit Connect™ platform that delivers subscription-based multi-domain research and analysis in the form of documents and publications delivered to a large volume of enterprise customers. Depending on subscription, clients have access to different volumes of content delivered through a dedicated web and mobile applications. One solution to the problem of finding relevant information is to split content into multiple pages based on extensive multi-level categorization, where each page is dedicated to a single subject (e.g. Brexit related research documents) or specific workflow, all supported by domain-specific content filtering capabilities. While this solution works really well in a workflow dedicated use cases, finding relevant documents can be improved by using personalized recommendation. To support this task, IHS Markit Connect™ platform developed a collaborative filtering-based recommendation system. In this paper we present the results of our work on improving the existing recommendation system using deep neural networks.

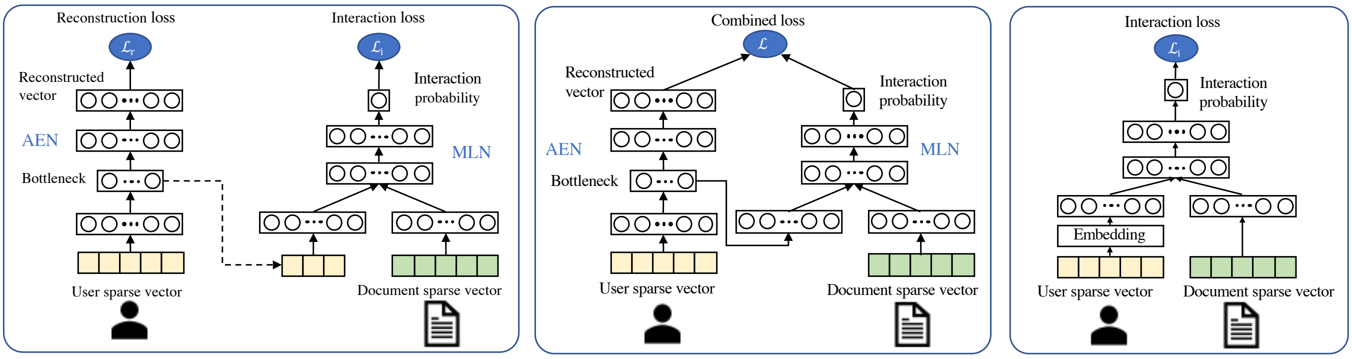
Several groups of methods are proposed to solve recommendation problem, including content-based filtering, collaborative filtering and hybrid methods. For the last couple of years deep learning architectures have been successfully applied in recommendation systems as an extension and integration of previously used methods and have become the new state-of-the-art.

Subscription-limited document recommendation can be seen as a special case of online news recommendation, a topic that is the subject of active research [1]. Recommending news, or generally documents in an online service is more challenging than recommending other types of items like movies in an online streaming service or products in an internet store due to the following challenges. Firstly, several new, most relevant documents are added or substituted every day which makes methods with a cold-start problem less effective. It also poses a challenge for models based on a fixed number of recommended items (e.g. the need for a frequent model update). Secondly, documents are time sensitive and their relevance drops quickly within a short period of time which makes the use of recommendation based on item popularity less effective. Thirdly, users' interests are time dependent and can be either periodic or can change over time which makes utilizing users' profiles for recommendation more challenging. All above problems affect effectiveness or increase the operational cost of traditional methods, like collaborative filtering or content-based filtering, and justifies the use of hybrid approach for recommendation, including deep learning-based hybrid methods.

To address the challenges of documents recommendation and to improve the performance of the existing recommender we propose a hybrid deep neural network architecture based on autoencoder (AE) and multilayer perceptron (MLP) for a subscription-limited document recommendation task. Our approach is based on the following assumptions:

- Time-limited user-document interactions statistics are sufficient to capture user similarity (i.e. users will have similar interaction statistics in many different, but long enough, periods of time)
- Existing document attribution is sufficient to capture the diversity of document content

The first assumption can be justified by the fact that a subscription limits the diversity of the content a user is entitled to and the fact that the content interaction is work-related, hence similarities between users do not change significantly over time. The second assumption can be justified by the extensive and diverse, manually assigned set of categories for each document.




a) Separate model b) Joint model c) Embedding model

Fig. 1. Visualizations all evaluated model architectures

During training we first learn a dense user representation using a sparse user interaction history and AE network. Then, using MLP network and taking a dense user embedding and sparse document feature vector as inputs we learn to predict the likelihood of user-document interaction. We consider 2 architectures: separately learned components and a combined model which archives better results. We compare the performance of our models and their variants with the existing recommender system and selected baselines. Experiments show significant performance improvement in comparison to the existing model. We plan to release our new proposed recommendation system in IHS Markit Connect™ initially as a dedicated page and widget on the mobile and web applications, using push notifications and later in many other document-related user activities.

The contributions of our paper are as follows:

- We introduce a novel dataset based on subscription-limited document interaction and extensive document attribution
- We propose the use of a model combining AE and MLP for document recommendation
- We evaluate the performance of deep learning models on novel dataset

## II. Our Approach

### A. Problem formulation

We formulate the document recommendation task using implicit feedback [2] and using the following sources of interactions: web application, mobile app, push and email notifications. We denote the set of users as $U = \{u_1, u_2, \dots, u_{|U|}\}$ and the set of documents as $V = \{v_1, v_2, \dots, v_{|V|}\}$, which are documents that were opened by any user from $U$ during a predefined period of time (e.g. 6 months). We define $U$-by- $V$ binary interactions matrix $I \in \mathbb{R}^{|U|\times|V|}$. In our dataset $I_{uv} = 1$ if a user $u$ opened a document $v$ during the predefined time period and $I_{uv} = 0$ otherwise. The binary vector $I_{u*} \in \mathbb{R}^{|V|}$ is a sparse representation of the user $u$. We denote document features as $F = [f_1, f_2, \dots, f_{|F|}]$ (e.g. geography, industry, sector) and interaction context features as $C = [c_1, c_2, \dots, c_{|C|}]$ (e.g. age of document or day of a week during interaction).

The goal of the recommender system is to predict the probability of interaction between the user $u$ and a candidate document $v$ at time $t$ given user representation $I_{u*}$, document representation $F_v$ and interaction information $C_{vt}$.

### B. Model architecture

In our model we aim to learn both user-to-user and user-to-document similarities. To this end, we propose recommendation system composed of two neural network components: AEN and MLN as in Fig. 1 a and b. We evaluate two variants of our architecture: components trained separately (Separate model) and components trained jointly (Joint model).

The AEN network component is a stacked autoencoder [3] used to reduce the dimensionality of user interaction vector $I_{u*}$ as well as to learn the latent, dense representation of each user. Formally, we can define an autoencoder as a neural network which consists of two functions: encoder $H$ and decoder $G$. In order to obtain the latent representation of an input vector we are minimizing loss function $L(x, G(H(x)))$. In our case we are only interested in the output of encoder part $H(x)$ which is usually described in literature as bottleneck. Our AEN network is composed of 2 hidden layers in encoder and 2 hidden layers in decoder and uses "rectified linear unit" (ReLU) [5] nonlinearity. During training we use binary cross entropy loss with weights to compensate sparsity of user-document interactions. In Separate model we consider 3 variants of the AEN network: vanilla autoencoder, denoising autoencoder [6] and autoencoder with dropout applied to hidden layers in Separate model. In Joint model we use vanilla autoencoder.

The MLN network component uses two heads: first taking a dense user vector from a bottleneck layer of the AE network, then taking the concatenation of a candidate document vector and interaction information vector and is 5-layer MLP neural network with SELU nonlinearity in all hidden layers. We train the MLN network using binary cross entropy loss.

## III. Related Work

### A. Recommender systems overview

There are two general approaches for building recommender systems: collaborative filtering (CF) and content-based filtering (CB). CF is based on finding similarities between users by looking at past user-item interactions (ratings or clicks) and omitting features of recommended items. CB methods find items similar to those accessed in the past. To address shortcomings of both methods (the cold-start problem for CF and the lack of diverse recommendations for CB) hybrid approaches, combining CF and CB methods, were developed. More recently, deep neural networks have become the new state-of-the-art of the recommender systems [1] archiving better results than previously used methods like matrix factorization or k-nearest neighbors algorithm.

## B. Autoencoder and news recommenders

The autoencoder neural network architecture is widely used in recommender systems [1]. For example, Cao et al. in [7] implement CF system by using a stacked autoencoder to compress user and item vectors and then by using cosine similarity between compressed vectors compute the probability of user-item interaction. A number of researchers use variants of autoencoder in a hybrid recommender system, including SVD++ model in [8], which uses contractive autoencoder [9] as a building block.

Interestingly, it is uncommon to use autoencoders in deep learning approaches for news recommendation [1]. More commonly, user embedding is created using the sequence of documents from a user's history. In the process of creating the user representation, first document embedding is created using features extracted from the document (e.g. using title) and then the sequence of document representations is compressed into the user embedding by using sequence models like LSTM [10] or Transformers [11]. For example, Zhu et al. in [12] use news embeddings created from titles with Convolutional Neural Network (CNN) as an input to LSTM network with an attention mechanism to create „user history sequential embedding". Wu et. al [13] use a multi-head self-attention network [11] to learn news representations from document titles and to learn user embedding.

# IV. Dataset, experiments and results

## A. Dataset information and statistics

We have built our dataset using historical user-document interactions in the IHS Markit Connect™ platform. We have used 6 months of interactions for training and 1 month of data for the evaluation of performance. Our training data consists of 155352 documents, 46531 users and 2666252 interactions. There are various usage patterns and the dataset covers both new documents that are read frequently and historical research that is accessed occasionally. The most popular document was read 15814 times, but at the same time 38% of all documents in the dataset were read only once (see Fig. 2). The most active user accessed documents 9445 times and only 13% of all users in the dataset accessed at least 100 documents (see Fig. 3).

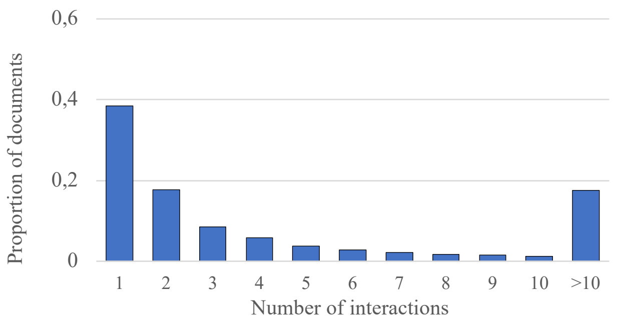



Fig. 2. Distribution of the number of document interactions

Documents are assigned shared categorical attributes: Domain (Energy, Chemical, Economics etc.), Type of Content (e.g. Report, Headline Analysis, Insight, Newswire), Geographical Location and domain-specific attributes like Chemical Product, Company or Industry. All categorical attributes are multivalued and can have from 7 up to 7203 items. For example, Geographical Location attribute has 382 possible values and 35% of all documents in the training set are assigned one location, 10% of documents - two locations and more than 1500 large, multi-chapter documents are assigned 54 locations. All attributes are hand-picked by document authors as part of the editorial process.

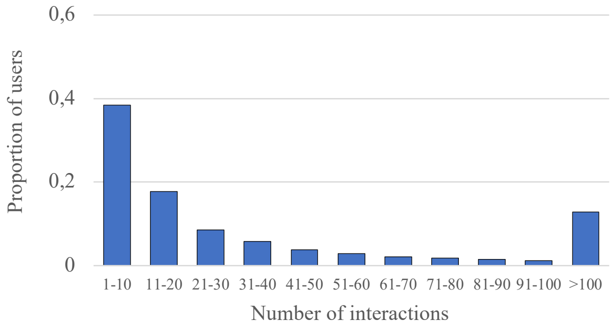



Fig. 3. Distribution of the number of user interactions

For our experiments we picked 16 document features: 15 categorical attributes and a document's estimated reading time, measured in minutes. We added one interaction context attribute: a document's age as a difference between the document interaction and document creation time. Our candidate document feature vector has a total length of 10712.

## B. Experiments details

We have used interactions between August 2018 and January 2019 as positive training examples and add randomly sampled documents as negative examples. Our test data is based on historical interactions from March 2019. We use subscription entitlements to narrow down documents each user has access to.

During training the interaction date from positive examples was used as the recommendation date and used to calculate a document's age. Negative examples were randomly sampled from all user-entitled, unread documents. Our interaction matrix $I$ and user vector $I_{u*}$ was built using documents with more than 5 interactions. Our experiments were performed using DGX servers.

## C. Evaluation procedure and metrics

During testing we score and rank all user-entitled documents published 2 weeks prior to a fixed recommendation date and we take all interactions collected for 1 day and for 1 week from the recommendation date. We use interactions found in top 2, 5, 10, 20 and 50 of ranked documents to measure recommendation performance using *Hits* and *Hit Rate* metrics.

## D. Model variants and baselines

We evaluate our Separate and Joint models and compare their performance against selected baselines. With the Separate model we evaluate 3 autoencoder variants: vanilla (S+V), denoising (S+N) and with dropout (S+D). Denoising autoencoder uses input masking with probability of 60%. Our Joint model (JNT) uses vanilla autoencoder and we construct loss function as a weighted sum of AEN network reconstruction loss and MLN loss with 1:2 ratio.

We compare performance of our models with 3 baselines: the existing recommendation system (EXI), embedding model (EMB) and ranking based on document popularity (POP). The currently running system is a collaborative filtering recommender using Jaccard index to calculate the similarity between users and it is boosted using the document's age. Boosting coefficients were determined empirically using historical interactions for each type of documents. The embedding model is a variant of the MLN model but uses the embedding layer to compress user representation (see Fig. 1 c). The document popularity baseline uses the number of document's reads for ranking.

### *E. Results and analysis*

TABLE I. PERFORMANCE COMPARSION USING HITS METRIC

| | S+V | S+N | S+D | JNT | EXI | EMB | POP |
|---|---|---|---|---|---|---|---|
| | | | | *1-day* | | | |
| top-2 | 81 | 92 | **95** | 82 | 6 | 47 | 3 |
| top-5 | 210 | 204 | 208 | **221** | 13 | 166 | 9 |
| top-10 | 313 | 312 | 320 | **370** | 24 | 337 | 15 |
| top-20 | 467 | 476 | 475 | **571** | 38 | 493 | 31 |
| top-50 | 715 | 718 | 694 | **843** | 78 | 737 | 65 |
| | | | | *1-week* | | | |
| top-2 | 270 | **278** | 269 | 254 | 44 | 186 | 7 |
| top-5 | **696** | 657 | 655 | 653 | 96 | 556 | 22 |
| top-10 | 1070 | 1084 | 1060 | **1152** | 165 | 1096 | 43 |
| top-20 | 1609 | 1631 | 1609 | **1815** | 263 | 1651 | 128 |
| top-50 | 2379 | 2377 | 2349 | **2639** | 498 | 2411 | 282 |

TABLE II. PERFORMANCE COMPARSION USING HIT RATE METRIC, ALL VALUES ARE PERCENTS

| | S+V | S+N | S+D | JNT | EXI | EMB | POP |
|---|---|---|---|---|---|---|---|
| | | | | *1-day* | | | |
| top-2 | 1.08 | 1.23 | **1.26** | 1.09 | 0.12 | 0.62 | 0.04 |
| top-5 | 2.80 | 2.71 | 2.76 | **2.93** | 0.26 | 2.21 | 0.11 |
| top-10 | 4.16 | 4.15 | 4.25 | **4.92** | 0.48 | 4.48 | 0.18 |
| top-20 | 6.20 | 6.32 | 6.31 | **7.59** | 0.76 | 6.55 | 0.38 |
| top-50 | 9.51 | 9.54 | 9.22 | **11.20** | 1.57 | 9.80 | 0.79 |

Table 1 and Table 2 show the performance comparison of all variants of our model and baselines using *Hits* and *Hit Rate* metrics. All deep learning models significantly outperform the existing model collecting at least 13 times more hits in top-10 1-day recommendations than the existing recommender baseline. Using *Hit Rate* metric, deep learning models get a score between 4.15% and 4.92% in top-10 recommendations, while the existing recommender gets only 0.48%. The popularity-based ranking baseline gets 0.18% *Hit Rate* in top-10 recommendations and is the worst performing recommender in our tests.

All evaluated autoencoder variants achieve similar performance. This result is quite surprising and can be attributed to specifics of client-document interactions in the environment limited by entitlements.

Overall the best performing model is a jointly trained JNT network, it outperforms Separate models in most categories and EMB baseline in all categories.

There are two major factors contributing to the significant performance improvement of our models: the use of deep learning and inclusion of document features. With deep neural networks, recommenders are able to learn representations that allow modeling complex, non-linear dependencies between users. The inclusion of diverse set of a document's features along with document's age enables modelling document-to-document and user-to-document dependencies inside a neural network and additionally reinforces the recommendation performance. Indeed, during our experiments we noticed that existing system is biased toward recommending most popular documents and our deep learning models' recommendations used a more diverse set of signals. But since we utilize both user-to-user and user-to-document similarities in our hybrid recommendation system we do not know which of the two similarities has a bigger impact on the model's performance. Based on the performance analysis of EXI, EMD,S+D and JNT (see Fig. 4) models we hypothesize that in our domain very little information is used from user-to-user similarity and much of the performance improvement comes from paring users with documents.

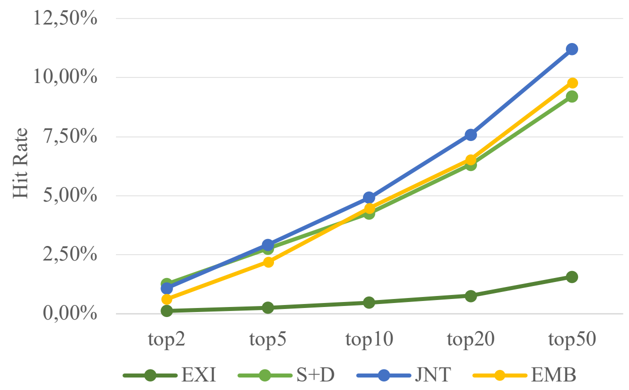



Fig. 4. Performance comparsion of selected deep learning models and the existing system

## V. CONCLUSION

In this work, we presented the problem of recommending subscription-limited document content in an online platform: IHS Markit Connect™. To improve the existing, CF-based recommender we proposed a deep learning model based on AE and MLP neural networks. Using historical interactions and document's categorical attributes we trained our models and showed that a new neural network-based, hybrid recommender significantly outperforms the existing system. We pick the best performing variant, Joint model, to replace the existing system on the platform. Using it, we will extend a recommender-based personalization in many document-related workflows and will conduct A/B testing against existing system to validate our synthetical test results. In the future we plan to continue improving recommendation performance by extending information we capture with implicit feedback, by using collected implicit negative feedback during training and by using different neural network architectures.

## ACKNOWLEDGMENT

This work has been partially supported by Statutory Funds of Electronics, Telecommunications and Informatics Faculty, Gdansk University of Technology.

## REFERENCES

[1] S. Zhang, L. Yao, A. Sun, "Deep learning based recommender system: A survey and new perspectives", vol. abs/1707. 07435, 2017-2019

[2] Y. Hu, Y. Koren, C. Volinsky, "Collaborative filtering for implicit feedback datasets", In ICDM,pages 263-272, 2008.

[3] P. Baldi, "Autoencoders, unsupervised learning and deep architectures" in Proc. ICUTLW, 2012, pp. 37–50.

[4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A simple way to prevent neural networks

from overfitting." J. Mach. Learn. Res., 15(1):1929–1958, January 2014

[5] V. Nair, G. E. Hinton, "Rectified linear units improve restricted boltzmann machines" in ICML, 2010

[6] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders" in ICML 2008

[7] S. Cao, N. Yang, and Z. Liu, "Online news recommender based on stacked auto-encoder" in Proceedings - 16th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2017, 2017, pp. 721–726.

[8] S. Zhang, L. Yao, X Xu, "AutoSVD++: An E.cient Hybrid Collaborative Filtering Model via Contractive Autoencoders", arXiv preprint arXiv: 1704.00551,2017

[9] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction" In Proceedings of the 28th international conference on machine learning (ICML-11). 833–840

[10] S. Hochreiter, J. Schmidhuber, "Long short-term memory", Neural Computation. 9 (8): 1735–1780, doi:10.1162/neco.1997.9.8.1735

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin. "Attention is all you need", Advances in Neural Information Processing Systems, pages 6000–6010.

[12] Q. Zhu, X. Zhou, Z. Song, J. Tan, L. Guo, "DAN: Deep attention neural network for news recommendation", in: AAAI 2019, 2019, pp. 5973–5980

[13] C. Wu, F. Wu, M. An, J. Huang, Y. Huang, X. Xie, "Neural News Recommendation with Attentive Multi-View Learning", arXiv preprint arXiv: 1907.05576,2019