User Modeling and User-Adapted Interaction
https://doi.org/10.1007/s11257-020-09282-4

Meta-User2Vec model for addressing the user and item cold-start problem in recommender systems

Joanna Misztal-Radecka[1,2] · **Bipin Indurkhya**[3] · **Aleksander Smywiński-Pohl**[1]

Received: 31 October 2019 / Accepted in revised form: 25 September 2020

Abstract

The cold-start scenario is a critical problem for recommendation systems, especially in dynamically changing domains such as online news services. In this research, we aim at addressing the cold-start situation by adapting an unsupervised neural User2Vec method to represent new users and articles in a multidimensional space. Toward this goal, we propose an extension of the Doc2Vec model that is capable of representing users with unknown history by building embeddings of their metadata labels along with item representations. We evaluate our proposed approach with respect to different parameter configurations on three real-world recommendation datasets with different characteristics. Our results show that this approach may be applied as an efficient alternative to the factorization machine-based method when the user and item metadata are used and hence can be applied in the cold-start scenario for both new users and new items. Additionally, as our solution represents the user and item labels in the same vector space, we can analyze the spatial relations among these labels to reveal latent interest features of the audience groups as well as possible data biases and disparities.

Keywords Recommender system · Neural embeddings · Cold-start · Doc2Vec model

✉ Joanna Misztal-Radecka
misztalradecka@agh.edu.pl

Bipin Indurkhya
bipin.indurkhya@uj.edu.pl

Aleksander Smywiński-Pohl
apohllo@agh.edu.pl

[1] AGH University of Science and Technology, Krakow, Poland

[2] Ringier Axel Springer Polska, Warsaw, Poland

[3] Cognitive Science Department, Institute of Philosophy, Jagiellonian University, Krakow, Poland

Published online: 20 October 2020

1 Introduction

This work aims to address the cold-start problem in recommendation systems for both the new-user and the new-item situations. We focus here on defining a strategy for the *complete cold-start* (CCS) situation when no historical browsing data are available [in contrast to *incomplete cold start* when a small number of records are available (Wei et al. 2016)]. Toward this goal, we propose Meta-User2Vec, which adapts the User2Vec method of Phi et al. (2016) based on the Doc2Vec architecture of Le and Mikolov (2014) to represent new users and new items in the recommendation system in an unsupervised way. More specifically, the work presented here makes the following contributions:

- We propose a hybrid Meta-User2Vec model, which enables modeling users and items with unknown history based on their metadata as described in Sect. 3 and can be applied for both the user and the item cold-start problems.
- We evaluate our proposed approach with respect to different model architectures and parameters and compare its performance with a factorization machine approach on three real-world datasets (Sect. 4). In Sect. 5, we summarize the results and possible applications of this approach.
- As our solution builds both the user and the item metadata embeddings in the same vector space, it allows analyzing underlying latent correlations, thereby revealing hidden characteristics of the audience and potential data biases.

To the best of our knowledge, there is no existing research that fully exploits the potential of the Doc2Vec model in terms of modeling both new users and new items, as discussed in Sect. 2. Moreover, as we use a popular unsupervised Doc2Vec architecture, our solution is a useful alternative to more complex models for real-world applications. Finally, we indicate further research directions in Sect. 6.

2 Motivation and background

2.1 From word vectors to user embeddings

The collaborative filtering user-profiling problem may be represented analogously to an NLP task: the items can be considered as words in a corpus, the user profile as a document and the sequences of user actions (such as buying a product or reading an article) as sentences. Consequently, the text embedding approaches have also been successfully adapted to collaborative filtering methods.

The Word2Vec model proposed by Mikolov et al. (2013) is one of the most popular unsupervised word embedding models that has been shown to be highly effective compared to the standard complex neural models while being computationally efficient. To represent whole texts as numerical vectors, some

researchers including Mikolov et al. (2013); Wang et al. (2018); Conneau et al. (2018); Arora et al. (2016) use a simple approach that averages all the document word vectors, thereby providing a strong baseline for many NLP tasks. In Le and Mikolov (2014), an extension of Word2Vec, Paragraph Vector (Doc2Vec) is proposed to represent text as a fixed-size dense vector, while learning semantic relations between words in parallel. It was found by Dai et al. (2015); Lau and Baldwin (2016) that Doc2Vec outperforms the other approaches, including more complex ones. It was also observed that the simpler PV-DBOW (Paragraph Vector—Distributed Bag of Words) version of Doc2Vec, which ignores the word order, is superior to PV-DM (Paragraph Vector—Distributed Memory) and that using pre-trained word embeddings for initializing document vector training improves the performance. Additionally, this study found that Doc2Vec performs better for longer texts, while short paragraphs are better modeled by vector averaging.

Word2Vec and Doc2Vec techniques have proved to be competitive alternatives to the traditional matrix factorization methods in the context of collaborative filtering recommendation systems and, due to its relative simplicity and efficiency, have been successfully applied for real-world recommender systems in different domains as described by McCormick (2018) such as venues in Ozsoy (2016); Grbovic (2018), e-commerce in Phi et al. (2016); Grbovic et al. (2016) and music in Barkan et al. (2016); Karam (2017). However, it was found by Caselles-Dupré et al. (2018) that the optimal parameters of the Word2Vec model for recommendation tasks are significantly different than for NLP. Moreover, there have been a few attempts to apply text embedding methods to represent content-based user profiles. In Musto et al. (2016); Alekseev et al. (2017); Misztal-Radecka (2018), Word2Vec is used to build content-based user profiles. It was observed that this approach gives comparable results to the standard collaborative filtering techniques, especially for sparse datasets.

2.2 User modeling in cold-start situations

The *cold-start* problem may be illustrated with two typical scenarios for an online news service. In the first one, a user, whose browsing history reveals the interest in football, visits a website during the World Cup and is shown fresh news from the latest competitions. In the second situation, a technology enthusiast, who uses incognito mode in the browser, thereby hiding any previous interactions from the recommender system, is shown recommendations of new gadget reviews.

In the first example, we are interested in showing items relevant to the user's interests, without having any information about the interaction history for new articles (*item cold start*). The online services provide a vast number of resources, but only a few items are read by any particular visitor. Due to the dynamic nature of the content—fresh articles are generated regularly at a rapid rate—there is often no click data for most items, so some content-based strategy is required to address this issue. Therefore to retrieve items relevant to the user's preferences, it is essential

Table 1 A summary of related works on the cold-start problem and the settings in which they were evaluated—new user, new item and if the complete cold-start (CCS) problem was considered

Reference	Model	New user	New item	CCS
LightFM Kula (2015)	FM	X	X	X
ECF Zhou et al. (2017)	Word2Vec	X	–	–
Meta-Prod2Vec Vasile et al. (2016)	Word2Vec	–	X	–
CHAMELEON de Souza Pereira Moreira et al. (2018)	RNN	–	X	X
CTM Wang et al. (2011)	LDA+MF	X	–	X

to find a meaningful representation of their behavioral profiles considering latent semantic features of the items in their browsing history.

In the second case (*user cold-start* problem), to prepare a personalized version of a given website and a recommendations list for first-time visitors, a strategy is required to make an initial guess about their interests and to find a group of like-minded users. This scenario also occurs when, due to privacy protection, cookie removal and the use of *incognito* mode, the browsing history of a user is unknown. One approach is to use other similarity measures considering the user metadata rather than behavioral patterns. It was observed by Goel et al. (2012); Misztal-Radecka (2018) that there are some significant differences in the frequency of interaction with relevant categories of online services among different demographic groups. Though relevant, the demographics information is usually not available for a majority of new users and some other features are required to predict user preferences. In Xu et al. (2011), differences in the use of mobile apps were analyzed, and it was found that the type of device, among other factors, influences user behavior.

We propose here Meta-User2Vec to represent new users and items in the recommendation system in an unsupervised way by building their metadata embeddings. Table 1 summarizes the related work with respect to the type of cold-start problem that is addressed. The intuition behind our idea is reminiscent of the LightFM model proposed by Kula (2015), which we use as a baseline for the experimental validation. Both approaches aim at learning user and item metadata embeddings for improving the recommendations in the cold-start setting; however, LightFM is an extended version of the matrix factorization method, while our work is based on the neural Doc2Vec model of Mikolov et al. (2013). In this context, our work shares a common ground with Meta-Prod2Vec of Vasile et al. (2016), which extends the Prod2Vec method of Grbovic et al. (2016) by leveraging user interactions with items and their attributes as side information within the Word2Vec architecture proposed by Mikolov et al. (2013). Both solutions are easy to implement as they do not require any modification of the source code of the original embedding model and only require operations on the input data. However, Meta-Prod2Vec considers only the item metadata for representing new items, while in our approach, user metadata vectors are trained along with item textual embedding, thus addressing both the new-user and the new-item situations. Additionally, in Vasile et al. (2016), the item

metadata are used only during the training phase; hence, only the incomplete cold-start situation is considered.

Another work that tackles the new-user problem with the use of Word2Vec embeddings is the embedded collaborative filtering (ECF) approach proposed by Zhou et al. (2017). It was shown to outperform the other state-of-the-art techniques in the cold-start situation for movies and retail data sets. However, this approach is limited to the incomplete cold-start problem, whereas we focus on representing users with no previous history. Another related approach is the Context2Vec model proposed by Stiebellehner et al. (2018) for look-alike modeling and recommendations on mobile apps. They represented user profiles as Doc2Vec embeddings from a concatenation of item descriptions and additional contextual features such as the operating system and the city, and item metadata to find that incorporating additional metadata in Doc2Vec model leads to improvement of classification measures. However, in this approach, the metadata are concatenated with item descriptions, whereas we use it as a separate input to the Doc2Vec to provide a mapping between their embeddings. Moreover, in contrast to our research, only a qualitative evaluation of this method was presented for the user-item recommendation task, and in Phi et al. (2016), the Doc2Vec method was only evaluated in the warm-start situation.

Apart from Word2Vec applications, several solutions employing deep learning methods for jointly modeling the user and the item metadata were published (Zheng et al. 2017; de Souza Pereira Moreira et al. 2018; Kumar et al. 2017). The CHAMELEON system (de Souza Pereira Moreira et al., 2018) is related to our project as it also provides a solution for the cold-start problem in news recommendation. However, it is based on a deep learning RNN architecture, whereas in our research, following the observations by Dacrema et al. (2019); Caselles-Dupré et al. (2018), we adapted a simpler Doc2Vec model. Moreover, de Souza Pereira Moreira et al. (2018) considers the next article prediction task, which is not applicable in the new-user setting. In Wang et al. (2011), the authors introduced a collaborative topic modeling technique that combines the collaborating-filtering approach with the content-based features extracted by topic modeling. However, this approach does not consider the new-user situation either.

Finally, we are interested in the "extreme cold-start" situation when both the user and the item history are unknown and we need to build a bridge between the user and the item metadata representations. The CB2CF model proposed by Barkan et al. (2019, 2016) introduces a similar concept of building a mapping between the content-based and the CF item embeddings. This neural model takes as input item descriptions (as text) and metadata and predicts CF latent embedding vectors, whereas we aim to build a mapping of user metadata into both CF and CB feature space.

3 Proposed approach

Our approach is based on the unsupervised Doc2Vec model of Le and Mikolov (2014) where the users are represented analogically to sentences, with word sequences replaced by clicked-item labels from their history.

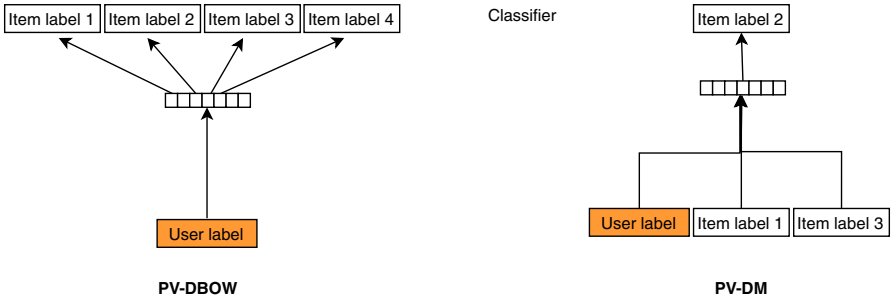

Fig. 1 Meta-User2Vec architecture with PV-DBOW and PV-DM Doc2Vec variants: Paragraph Vector DBOW (left) and DM (right), adapted from Le and Mikolov (2014)

In the original Doc2Vec model, at each iteration of stochastic gradient descent, a text window is sampled, and a random word from this window is selected to form a classification task given the document vector. However, the label annotating a document does not need to be single or unique and may include additional information such as tags and other document metadata.

While the original Doc2Vec model learns to represent text as fixed-size dense vectors, in the User2Vec approach to recommendations, the user-id embedding vectors are trained based on the sequences of items in their history (Phi et al. 2016). We propose an extension of the basic User2Vec model: Meta-User2Vec method, which takes an additional input of user and item metadata labels to be trained along with the id vectors, which enables inferring these features for new items and new users. This idea may be seen as a generalization of Meta-Prod2Vec to represent both new users and new items within Doc2Vec architecture. Each element of the user metadata, such as demography, location, device, software version, and browser, as well as the user's identifier, is treated as a label. "User" is described by a sequence of metadata labels, which are input to a Doc2Vec model. The learning process in this approach builds a vector representation of the user metadata labels (such as device types and gender) so that groups of users who act similarly have spatially close representations. Moreover, these representations can handle similarity queries between the user and item-metadata labels, as well as between particular user ids and item ids, so that recommendation lists for particular user groups can be generated. When the user and the item metadata contain only id labels, the model is equivalent to the User2Vec method.

To define the problem more formally, we introduce the following notation:

- $\{u_1, u_2, \ldots u_N\} \in U$—the set of N users in the recommendation dataset,
- $\{a_1, a_2, \ldots a_L\} \in A$—the set of all L items in the recommendation dataset,
- $\{m_u^1, m_u^2, \ldots, m_u^k\}, u \in U, m \in M_U$—the set of labels (metadata or id) from the set of user metadata M_U for user u,
- $\{k_a^1, k_a^2, \ldots, k_a^d\}, a \in A, k \in K_A$—the set of labels (metadata or id) from the set of item metadata K_A for item a,
- $p_u \in \mathbb{R}^D$ – D-dimensional latent representation of user u,

- $q_a \in \mathbb{R}^D$ – D-dimensional latent representation of item a.

We consider two types of the Doc2Vec model architectures proposed in the original work of Le and Mikolov (2014) and adapt them to the recommendation task as presented in Fig. 1. To illustrate the idea behind both methods, let us introduce a persona—Alice, a student who liked the movies *Toy Story* and *Aladdin*.

- PV-DM (Fig. 1—right)—the paragraph vector and sampled context word vectors are averaged or concatenated to predict the center word from the given context.

  In the recommendation setting, we use the average of user-label and item-label vectors from the user history context to maximize the probability of the center item label—for example, maximizing the probability that Alice liked *Toy Story,* given that she is a student and liked *Aladdin*. The cost function is defined as:

  $J = -log(P(k_a^c | m_u^i, k_a^{c-w}, k_a^{c-w+1}, ..., k_a^{c+w}))$, where c is the index of the central item, w is the window size and i is the user's metadata index.
- PV-DBOW (Fig. 1—left)—the paragraph vectors are trained to predict words in a small window and may be interpreted as context vectors for the sampled text.

  For the recommendation setting, the user-label vector is trained to predict item labels within a sampled window, and the user label may be treated as the interaction context. For example, the model maximizes the probability that Alice likes the movies *Toy Story* and *Aladdin,* given that she is a student, by minimizing the following cost function:

  $J = -log(P(k_a^{c-w}, k_a^{c-w+1}, ..., k_a^{c+w} | m_u^i))$

  We use this method in combination with Skip-Gram pre-training of Word2Vec Mikolov et al. (2013) item vectors to maximize the log-likelihood of an item given its context—for example, the probability that Alice liked the movie *Toy Story* given that she liked *Aladdin*. The cost is given by:

  $J = -log(P(k_a^{c-w}, k_a^{c-w+1}, ..., k_a^{c+w} | k_a^c))$

  so that the resulting user p_u and item q_a embeddings are represented in the same latent space.

The user and item embeddings are calculated as follows:

- For a user u or item a available in the training set—use the embedding p_u or q_a calculated during the training process.
- For a new item a—use the item-metadata labels $\{k_a^1, k_a^2, ..., k_a^d\}$ to infer the item vector from the model.
- For a new user—calculate the user-embedding vector p_u as the average of user-metadata embeddings $p_u = \frac{1}{k} \sum_{i=1}^{k} p_{m_u^i}$.

As in both PV-DM (with vectors averaging) and DBOW (with skip-gram item vectors pre-training), the user- and item-label vectors are represented in the same space, we calculate the scores as cosine distance between the user and the item vectors:

$$score_{u,a_k} = \frac{p_u \cdot q_{a_k}}{\|p_u\| \cdot \|q_{a_k}\|}$$

Table 2 Meta-User2Vec configurations used in the experiments

Architecture	Window size	NS exponent	NS size	Iterations	Alpha
PV-DM (vector averaging) DBOW (skip-gram pre-training)	[5, 10, 20, 50]	[− 1, − 0.5, 0, 0.5, 1]	5	50	0.0025

The recommendations are generated as those items that have the most similar vectors q_a to the user representation p_u.

The Meta-User2Vec method is applicable for modeling interaction sequences and may be used as an alternative for standard matrix factorization-based methods in cold-start situations, especially when both user and item metadata are available. However, if the recommendation task is to predict non-binary user actions (such as the product rating), it should be combined with another method (such as user- or item-based kNN model).

4 Experimental validation

In the experimental evaluation, we aim at answering the following research question:

1. What is the impact of model architecture, hyper-parameters and input features in each situation?
2. How does the proposed method perform compared to other approaches and the baselines in each situation?
3. Is the proposed Meta-User2Vec architecture capable of representing latent relations among user and item metadata, and detecting potential data biases?

To address these questions, we performed the following five experimental tasks on three real-world datasets. As the proposed method is applicable in both warm and various cold-start settings, we conducted experiments in each configuration.

4.1 Models used for comparison

In the experimental evaluation, we compared the following models:

- Meta-User2Vec—Our proposed method described in Sect. 3. We compared two versions of the Doc2Vec architecture (PV-DBOW and PV-DM). The tested hyper-parameters for Meta-User2Vec model are presented in Table 2. As noted by Caselles-Dupré et al. (2018), the optimal hyper-parameter selection for the Word2Vec model applied to recommendations differs from the original NLP setting. One key parameter identified in this study was the negative sampling (NS) exponent, which defines the probability distribution of the items sampled in the negative sampling process. If this parameter is equal to 1.0, the sampling is pro-

portional to the word frequencies; if it is 0.0, all words are sampled equally; and if it is negative, low-frequency words are sampled with a higher probability. We perform an analogical comparison of different NS exponents and window size values for the Doc2Vec model applied to recommendations.

- LightFM Kula (2015)—A hybrid matrix factorization model representing users and items as linear combinations of the latent factors of their content features. The probability $\hat{r}_{ua}$ of interaction between a user u and an item a is modeled as the sigmoid of the dot product of the user vector and the item vector, along with the bias terms associated with the user and the item: $\hat{r}_{ua} = \sigma(q_a^T p_u) + b_u + b_a$. We selected this model for comparing with Meta-User2Vec for two reasons. First, it has been shown to outperform the classic matrix-factorization approaches. Second, it is a hybrid model that enables building representations on both the user and the item ids and metadata, and hence, it may be applied to the complete cold-start situation for both new items and new users.
- Non-negative matrix factorization (NMF) Lee and Seung (1999)—Finds a decomposition of user-item interaction matrix R into two matrices P and Q, with the user and the item latent features of non-negative elements, by optimizing the distance between the original matrix and the product of decomposed matrices. A reconstruction of the user-item interactions is performed by calculating $\hat{r}_{ua} = q_a^T p_u$, and the items with the highest predicted ratings are recommended for each user. We use this method as a baseline because it is an efficient and popular algorithm for collaborative filtering settings. However, as it builds the user and the item representations from the interaction matrix, it can be applied in the warm-start setting only.
- Random recommender—Randomly sorts out the available items in the test set. We apply this method as a baseline in all the recommendation settings.

We used 100-dimensional vectors for all the compared models (Meta-User2Vec, LightFM and NMF). For the hybrid models (Meta-User2Vec and LightFM), we compared the results when the user and the item metadata are used and for the id-labels only version (which cannot be applied in the cold-start situations). All the models were implemented in Python, using `gensim` (Řehůřek and Sojka 2010), Scikit-Learn and LightFM (Kula 2015) libraries.

4.2 Datasets

We used three real-world recommendation datasets that contain both user and item metadata: a popular public MovieLens 100K dataset, a public article sharing dataset from Deskdrop[1] and a private dataset from the Onet[2] news service. The detailed information about these datasets is presented in Table 3.

[1] https://www.kaggle.com/gspmoreira/articles-sharing-reading-from-cit-deskdro.

[2] www.onet.pl.

Table 3 Training data summary for MovieLens, Deskdrop and Onet recommendation datasets

	MovieLens	Deskdrop	Onet
Events range	7 months	1 year	2 weeks
Users count	942	1895	14,151
Avg positive interactions per user	58.8	12.4	21.3
Distinct items	1447	3047	11,252
User metadata	Gender, age, occupation	Geolocation, user–agent	Gender, age, user–agent
Item metadata	Title, genres, year	url	Text

4.2.1 MovieLens 100K dataset

A popular movie recommendation dataset was collected from the MovieLens movie rating website (Harper and Konstan 2015). This dataset contains users who gave at least 20 ratings and provided complete demographic information (gender, age and profession). The rating scale is 1–5, but as we are interested in predicting the items that the users would like, we binarize the movie dataset so that only items having high ratings within the user history (4 and above) are considered as positive records. For item metadata, we use the list of movie genres and the year of its release.

4.2.2 Deskdrop dataset

Dataset from CI&T's Internal Communication platform (Deskdrop) which enables the employees to share relevant articles with their peers, and collaborate around them. Different types of interactions were collected from logged-in users in different platforms (web browsers and mobile native apps). To binarize the problem and avoid the presentation bias during the evaluation, we used explicit actions (article like, share or comment) as positive labels and *view* event types as negative labels. We also used contextual information about user interactions (user–agent and geolocation). To construct the item metadata, the article URLs were split into alphanumeric expressions.

4.2.3 Onet dataset

The Onet dataset contains data from the Polish news service `www.onet.pl` and related websites of Ringier Axel Springer Polska for a sample of anonymous users who accepted the service cookie policy and the terms of use. Each record in the event table represents an interaction between a user and an item (when an article was clicked by a user). The outliers (users who made less than 5 and more than 100 clicks) were removed. The records in the history table might be accompanied by additional user metadata such as the device and the software type in a user–agent

string as well as declarative information about the user demographics, including gender (female—46% of users or male—54% users) and age. To avoid sparsity, users were divided into six age groups: under 20 (2% users), 21–30 (11% users), 31–40 (26% users), 41–50 (28% users), 51–60 (18% users), over 60 (15% users). We use the article texts as the item metadata. As a preprocessing step, words with fewer than 10 occurrences and stopwords based on a pre-defined list were removed. Next, the text was normalized to lowercase and words shorter than three characters and with non-alphabetic characters were filtered out.

The test data were collected over a period following the training period such that 90% of users and 66% of articles in the test set were not available during the training.

4.3 Experimental tasks

The detailed configurations of the experiments are described below. In the item or the user warm-start settings, we removed those records from the test set that were not present during the training.

1. **Warm start**—all the users and items from the test set are present in the training set.
   Split strategy: Random split on interaction matrix (75/25).
   $U_{test} = U_{train}, A_{test} = A_{train}$
   Datasets: MovieLens, Deskdrop
2. **Item cold start**—all the users from the test set are present in the training set, none of the items from the test set are in the training set.
   Split strategy: Random split on item ids set (75/25)
   $U_{test} = U_{train}, A_{train} \cap A_{test} = \emptyset$
   Datasets: MovieLens, Deskdrop
3. **User cold start**—all the items from the test set are present in the training set, none of the users from the test set are in the training set.
   Split strategy: Random split on user ids set (75/25)
   $U_{train} \cap U_{test} = \emptyset, A_{test} = A_{train}$
   Datasets: MovieLens, Deskdrop
4. **User and item cold start**—none of the users and items from the test set are in the training set.
   Split strategy: Random split on user and item id sets (75/25 for each)
   $U_{train} \cap U_{test} = \emptyset, A_{train} \cap A_{test} = \emptyset$
   Datasets: MovieLens, Deskdrop
5. **Next date**: split by date
   $max(date_{train}) < min(date_{test})$
   Dataset: Onet

4.3.1 Evaluation procedure

We treated recommendations as a binary problem of predicting user-item interaction probability. To reduce the presentation bias (when some items were not

Table 4 Best configuration in each experiment for MovieLens and Deskdrop datasets with respect to NDCG@5

Dataset	Experiment	Metadata	Model	Window	NS exp.
Deskdrop	Warm	Item: True, user: False	DBOW	50	1
	Cold item	User: True	DBOW	50	0.5
	Cold user	Item: True	PV-DM	50	– 1
	Cold user-item	–	DBOW	10	1
MovieLens	Warm	Item: False, user: True	DBOW	50	– 1
	Cold item	User: False	PV-DM	20	– 1
	Cold user	Item: True	PV-DM	5	– 1
	Cold user-item	–	PV-DM	5	– 1

viewed by the user), only the items that the user interacted with from the test set were considered during the evaluation. Following Kula (2015), for each of the experimental settings, we ran 10 iterations of random splitting and the results are reported as the average of test set results from all the runs. We used the ranking metrics—normalized discounted cumulative gain (NDCG) Wang et al. (2013) and precision (PREC) for top-5 recommended items as we argue that they are more appropriate than the binary metrics for the recommendation problem. PREC@5 measures the proportion of relevant item ids in the recommendation set and NDCG reflects the degree of the neighborhood (discount), which decreases with the distance. Mann–Whitney–Wilcoxon two-sided test with Bonferroni correction was used for calculating the statistical significance of the results with the following notation: No significance (ns): $0.05 < p \leq 1$; *: $0.01 < p \leq 0.05$; **: $0.001 < p \leq 0.01$; ***: $0.0001 < p \leq 0.001$.

As the user metadata representations are built along with the word vectors in the Meta-User2Vec model, and all the embeddings are in the same space, we explored the relations between the user and the item embeddings as well as the metadata tags to gain insights into their interaction patterns and to detect the hidden data biases. First, the vector representations of user metadata tags were mapped to a 2D space using Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) of McInnes et al. (2018), and the resulting spatial relations are explored for patterns. The resulting coordinates show the types of metadata representing similar user behaviors: neighboring tags describe users with similar browsing histories. Additionally, we explored lists of top-5 items that are closest to the generated metadata representations (using cosine distance), and qualitative analysis of the resulting recommendation list is carried out.

5 Results

Below, we present the results of the experiments with respect to the research questions posed in Sect. 4.

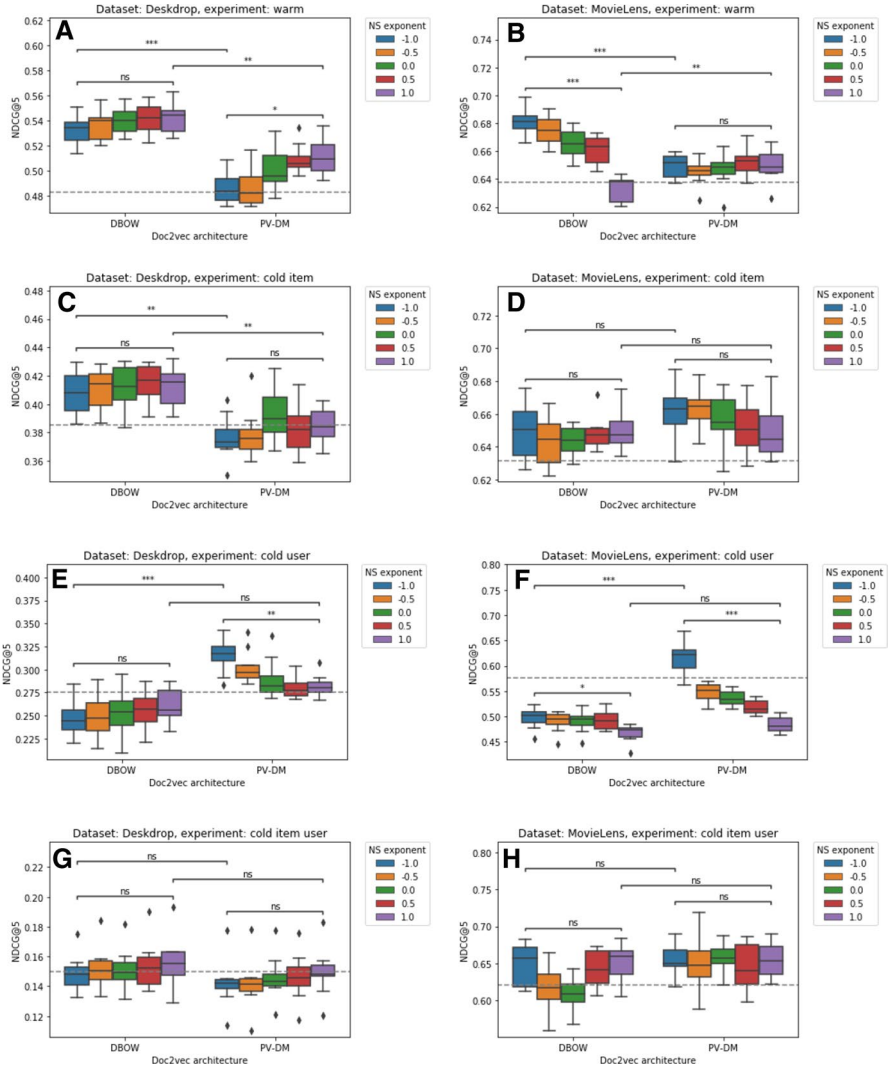

Fig. 2 Comparison of NDCG@5 for different architectures (DBOW vs. PV-DM) and parameters (negative sampling exponent) for the Meta-User2Vec model with user and item metadata for different experimental settings for Deskdrop and MovieLens datasets and statistical significance annotation between the extreme configurations. The dotted line represents the random baseline

5.1 The impact of model architecture, hyper-parameters and input features in each situation

As shown in Table 4, the best parameters for the Meta-User2Vec differ depending on the experimental setting and the dataset. In particular, for the Deskdrop dataset, the item features improve the performance in both new-item and warm-start

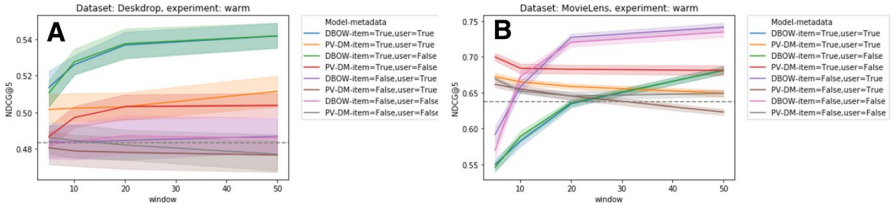

Fig. 3 Comparison of NDCG@5 for different architectures (DBOW vs. PV-DM) and parameters (window size) for the Meta-User2Vec model with user and item metadata for the user and item metadata labels for Deskdrop (**a**) and MovieLens (**b**) datasets in the warm-start experiment. The dotted line represents the random baseline

situations, which is not the case for MovieLens. The reason may be that in Deskdrop dataset, the interactions are more sparse and the item features (article URLs) may be more informative. On the other hand, for MovieLens, most of the items have a larger number of ratings while the item metadata (genres and publication year) are quite general. The user features do not have a significant impact on the results in any of the settings.

Figure 2 presents the comparison of results for different configurations of the Doc2Vec model in each experimental setting for the user and item metadata input, which may be applied in both cold-user and cold-item situations. In the warm-start setting, the DBOW model gives significantly (**) higher results than the PV-DM architecture for both datasets (Fig. 2a and b). Interestingly, the impact of the negative sampling distribution parameter depends on the dataset characteristics and architecture—for Deskdrop (Fig. 2a), positive values of NS exponent with the PV-DM architecture yield significantly higher results (*) than the negative ones. However, for MovieLens (Fig. 2b), the negative value NS exponent=-1 for DBOW gives significantly higher results (***) than the other configurations, while NS exponent=1 falls below the baseline. This difference may be related to the characteristics of both the datasets as the Deskdrop dataset is more sparse than MovieLens so that sampling more popular items may improve the results. Another factor may be the fact that for MovieLens the item labels are genres and release year, which are significantly less sparse than the textual data from the website URLs or text. Hence, modeling item labels based on textual content is more similar to a standard NLP task in terms of word distribution (the optimal value of NS exponent in Mikolov et al. (2013) was 0.75). As presented in Fig. 3a and b, another parameter that has a significant impact on the results is the window size—in both cases, the best results in the warm-start setting are achieved for the DBOW architecture with the largest window size 50. For the other experimental configurations, the window size did not have any significant impact.

For the cold item and cold item-user experiments, the differences among different negative sampling values (Fig. 2g and h) are not significant for both the datasets. For the Deskdrop dataset, the DBOW architecture yields better results (**) in the cold-item scenario (Fig. 2c), though PV-DM is better for MovieLens (Fig. 2d), but the difference is not significant.

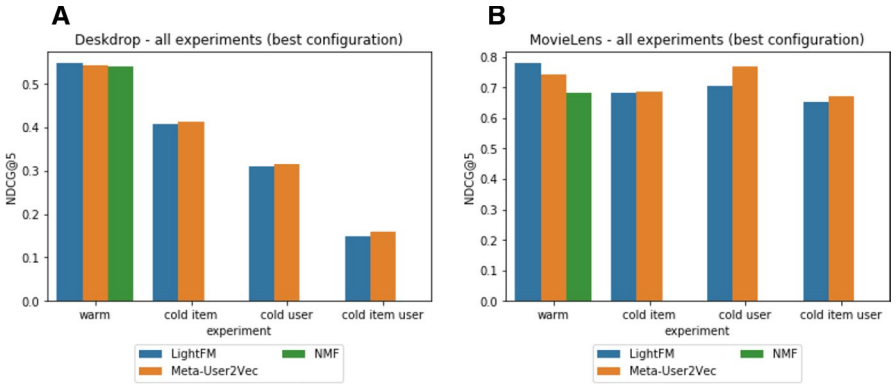

Fig. 4 Comparison of NDCG@5 for LightFM and Meta-User2Vec algorithms with user and item metadata for different experimental settings for Deskdrop and MovieLens datasets

In the cold-user experiment for both the datasets (Fig. 2e and f), the best results are achieved by the PV-DM model with NS exponent $= -1$ ($**$), while the DBOW model gives results below the baseline. This indicates that averaging the user and the item labels to predict the context word (PV-DM), combined with over-sampling low-frequency words (negative NS exponent), is a better strategy to model the user-item relations than predicting item metadata based on the user label, when only user metadata is available.

The analysis of the results for different configurations of the Doc2Vec architecture shows that the choice of parameters depends on the characteristics of the dataset—in situations when there are many new users in the test set, it may be beneficial to use a negative value of negative sampling exponent with a PV-DM architecture, and the choice of NS exponent depends on the dataset characteristics—for sparser interactions and metadata (such as texts) positive values should be selected as in the original NLP configuration, while for more dense datasets and labels, negative values may work better. Additionally, in the warm-start setting, DBOW benefits from a larger window size.

5.2 A comparison of the proposed method with other approaches and the baselines

Figure 4 summarizes the best results for each model for all the experimental settings. The detailed results of the compared models in all the experimental tasks with respect to the user and the item input feature combinations are presented in Fig. 5 and Table 5.

In the warm-start setting (Fig. 5a and b) when only the user-id and the item-id labels are available, the LightFM method yields significantly higher results than the Meta-User2Vec approach for both the datasets ($***$), and NMF is also significantly better than the id-based Meta-User2Vec for Deskdrop. However, when both user and item metadata are available, Meta-User2Vec gives significantly better results than LightFM

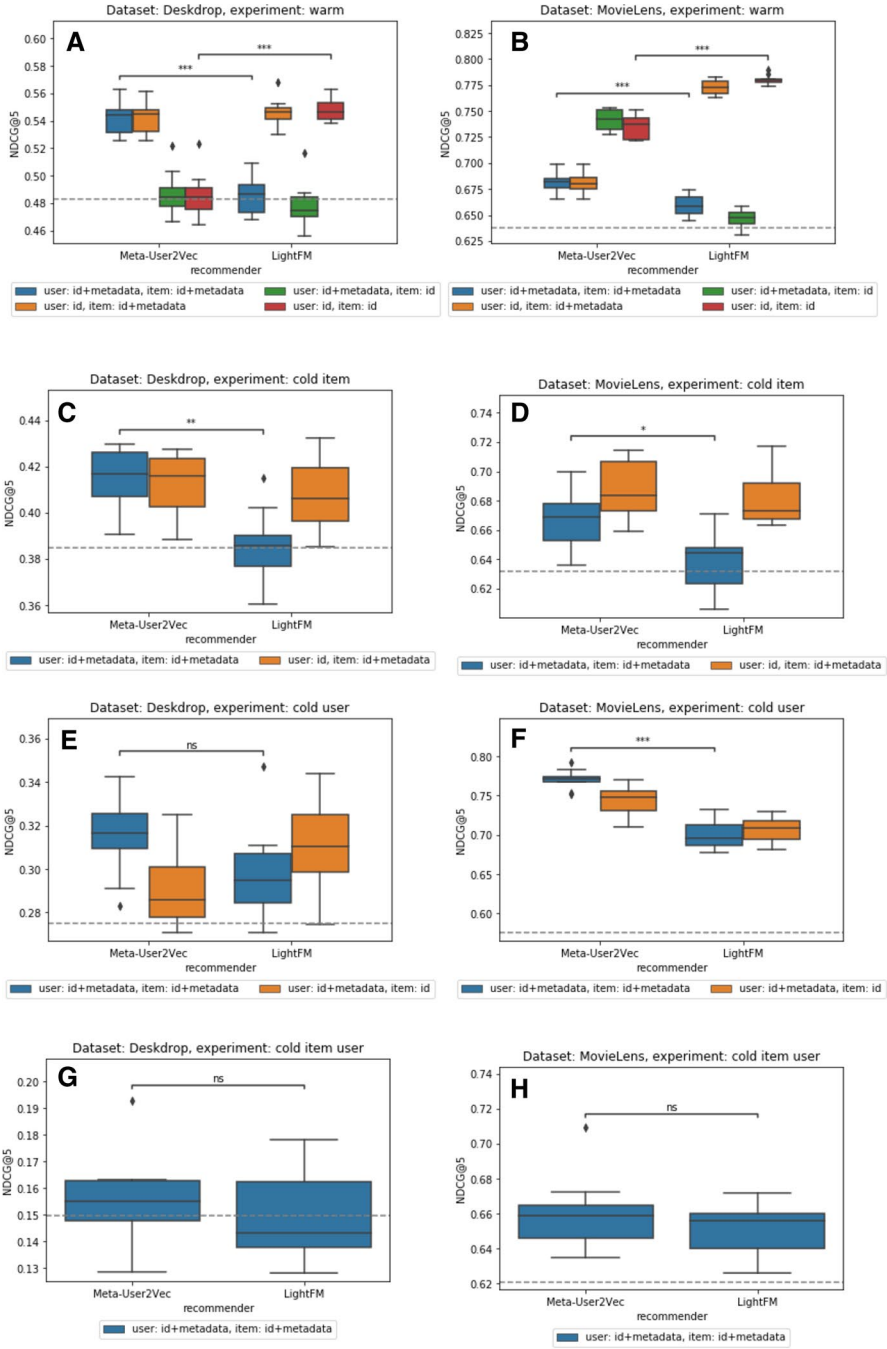

Fig. 5 Comparison of NDCG@5 for different recommendation algorithms (LightFM and Meta-User2Vec) and input features for different experimental settings for Deskdrop and MovieLens datasets with statistical significance annotation between the extreme configurations. The dotted lines represent the random baseline

Table 5 Comparison of NDCG@5 and PREC@5 for different recommendation algorithms (NMF, LightFM and Meta-User2Vec) and input features (with or without user and item metadata) for different experimental settings for Deskdrop and MovieLens datasets

Dataset	Experiment	Recommender	NDCG@5				PREC@5			
		User metadata	False		True		False		True	
		Item metadata	False	True	False	True	False	True	False	True
MovieLens	Warm	LightFM	**0.78**	0.773	0.647	0.659	**0.678**	0.672	0.562	0.572
		Meta-User2Vec	0.735	0.681	0.742	0.681	0.641	0.598	0.647	0.599
		NMF	0.672	–	–	–	0.586	–	–	–
		Random	0.637				0.555			
	Cold user	LightFM	–	–	0.707	0.700	–	–	0.698	0.688
		Meta-User2Vec	–	–	**0.753**	0.618	–	–	**0.744**	0.618
		Random	0.576				0.576			
	Cold item user	LightFM	–	–	–	0.652	–	–	–	0.577
		Meta-User2Vec	–	–	–	**0.657**	–	–	–	**0.586**
		Random	0.621				0.553			
	Cold item	LightFM	–	0.681	–	0.639	–	0.594	–	0.555
		Meta-User2Vec	–	**0.669**	–	0.663	–	**0.584**	–	0.577
		Random	0.631				0.549			

Table 5 (continued)

Dataset	Experiment	Recommender	NDCG@5				PREC@5			
		User metadata	False		True		False		True	
		Item metadata	False	True	False	True	False	True	False	True
Deskdrop	Warm	LightFM	**0.548**	0.546	0.478	0.485	**0.225**	0.225	0.193	0.198
		Meta-User2Vec	0.487	0.542	0.488	0.542	0.196	0.225	0.196	0.224
		NMF	0.537	–	–	–	0.22	–	–	–
		Random	0.485				0.197			
	Cold user	LightFM	–	–	0.31	0.297	–	–	0.183	0.177
		Meta-User2Vec	–	–	0.309	**0.315**	–	–	0.18	**0.187**
		Random	0.276				0.158			
	Cold item user	LightFM	–	–	–	0.149	–	–	–	0.062
		Meta-User2Vec	–	–	–	**0.156**	–	—	—	**0.065**
		Random	0.150				0.061			
	Cold item	LightFM	—	0.408	—	0.385	—	0.209	—	0.193
		Meta-User2Vec	—	0.413	—	**0.414**	—	0.215	—	**0.216**
		Random	0.383				0.193			
Onet	Next date	LightFM	—	—	—	**0.359**	—	—	—	**0.135**
		Meta-User2Vec	—	—	—	0.346	—	—	—	0.128
		Random	0.274				0.106			

The values in bold are the best results for each experimental settings with respect to each of the metrics

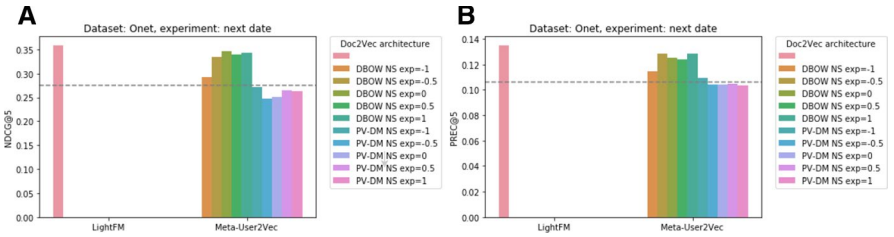

Fig. 6 Comparison of NDCG@5 (**a**) and PREC@5 (**b**) for Meta-User2Vec and LightFM for the next date experiment on Onet dataset. The dotted line represents the random baseline

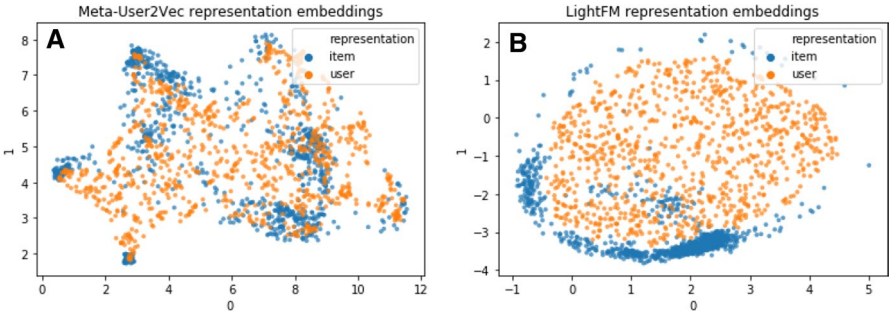

Fig. 7 User and item embeddings from the MovieLens dataset with the UMAP 2D mapping of Meta-User2Vec and LightFM models

in the same configuration. Interestingly, adding metadata does not always improve the model performance—for instance, in the warm-start case for MovieLens (Fig. 5b), both models trained only on the ids give better results than when metadata are added. The reason may be caused by the fact than incorporating metadata in the training process leads to modeling more general information than the case when the input is single ids. Though such generalization is helpful when not enough information is available about particular users and items, it may obscure available information about more active users (as in the case of MovieLens when only users with at least 5 ratings are considered).

Meta-User2Vec gives better results than LightFM for new-item situations (Fig. 5c, d, g and h). However, for the user-item scenario (Fig. 5g and h) the difference is not significant. In the cold-user experiment (Fig. 5c and d), the Meta-User2Vec model yields a lower performance than LightFM. In the next-date experiment on the Onet dataset (Fig. 6a and b), the DBOW model with NS exponent close to zero (all item labels are sampled equally) yields the best results for Meta-User2Vec, which is slightly worse than the results of the LightFM model.

5.3 Representing latent relations among user metadata and content

Figure 7 shows the embeddings of users and items in the two-dimensional space (transformed by the UMAP algorithm) for the Meta-User2Vec DBOW (Fig. 7a)

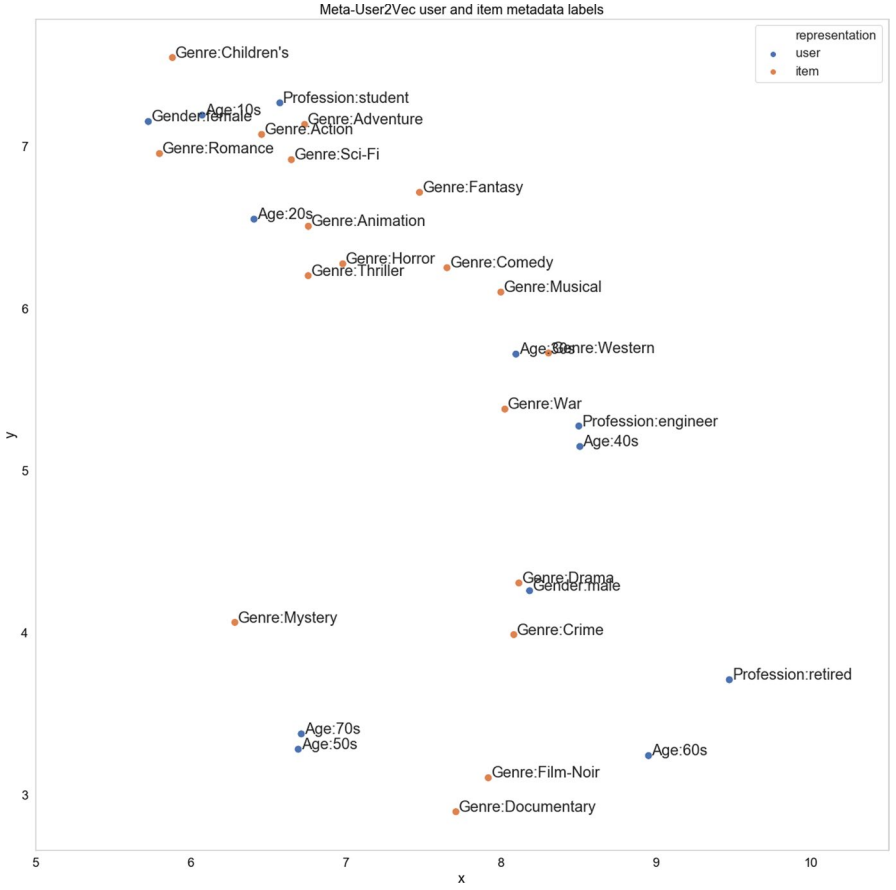

Fig. 8 UMAP 2D mapping of user and item metadata tags for the MovieLens dataset

and LightFM (Fig. 7b) models build with the user and item metadata inputs. We observe that the user and item embeddings built with Meta-User2Vec are within the same space, and the user representations are more widely distributed than the items, which seems intuitive as a user may be interested in different types of items. This observation indicates that Meta-User2Vec is capable of representing user-item relations, which may be useful for a qualitative analysis of the use patterns. To further explore this finding, in Fig. 8 we observe the spatial relations revealing some interesting interaction patterns among the user and the item features as well as possible biases from the MovieLens dataset. For instance, the *Gender:male* vector is closer to the representation of *Profession:engineer* and genres *drama*, *war* and *western*, whereas the vector representing the female users is close to *Genre:children's* and *Genre:romance* representation. The *Profession:student* is related to *adventure* and *action* movies, whereas the embeddings of the age groups *Age:50s*, *Age:60s* and *Age:70s* are close to the representations of *film-noir* and *documentaries*. The vector

Table 6 Examples of the closest item labels for selected user labels for different dataset

User label	Dataset	Recommendations
Gender:female	Onet	Lifestyle (fashion), celebrities, celebrities, lifestyle (fashion)
	MovieLens	Genre:Romance, *Dirty Dancing*, Genre:Drama, Genre:Children's, *Sense and Sensibility*
Gender:male	Onet	Sport (football), economy, sport, sport (hockey)
	MovieLens	Genre:Action, Genre:War, Genre:Adventure, Genre:Comedy, Genre:Crime
Age:60s	Onet	Politics, economy (retirement), politics, politics, politics
	MovieLens	Year:1934, Year:1944, Genre:War, Year:1952, Genre:Drama, Year:1955
OS:Vista	Onet	Lifestyle (home decoration), politics, lifestyle (cooking), lifestyle (cooking), religion
OS:Linux	Onet	Technology (smartphones), technology (smartphones), technology (smartphones), technology (smartphones), technology (Apple)
	Deskdrop	Technology (programming), technology (Linux), technology (Linux), technology (Android), technology (Github), technology (Android)
Device:Apple	Deskdrop	Technology (iOS), technology (Apple Pay), technology (Apple Pay), travel (airport), economy (Bitcoin), technology (Apple)

similarity may mean co-occurrences of user labels (for instance the label *Age:60s* and *Profession:retired*) but may also indicate behavioral patterns (for example when two different age groups are interested in similar items).

Similar observations hold for the other two datasets. The examples of the most similar item labels for particular user labels for all datasets are presented in Table 6. In particular, *female* vector is associated with fashion, beauty and celebrities (Onet) and romance movies such as *Dirty Dancing* (MovieLens), *male* is close to articles about sports and economy (Onet) and action and war movies (MovieLens), while the tag of *60+* lies close to politics, including articles about retirement (Onet) and old movies (MovieLens). Among the user–agent-based labels, *Linux* is close to technology and programming (Onet and Deskdrop), *Vista* to home and cooking (Onet), while *Apple* representation is related to articles about *iOS*, *Apple Pay* and other Apple products. These results show that our proposed method of representing user metadata is capable of capturing the latent behavioral patterns of the user groups.

These examples indicate that the resulting label similarities for the DBOW model show the most characteristic interest patterns for a particular user group rather than the most popular ones. While this behavior may be beneficial in the user-item recommendation scenario to find the most suitable items for a particular user, such model characteristics may lead to a lower performance of this method in the user cold-start scenario compared to the PV-DM version. Moreover, this approach may amplify the existing data biases and stereotypes. Therefore, depending on the application, the analysis of embedding similarities should be applied to identify such situations. We also recommend combining this approach with some exploration-enhancing mechanisms to prevent the filtering bubble effect.

5.4 Results summary and discussion

The comparison of models in each of the experimental settings shows that the choice of the model, as well as its architecture, is dependent on the characteristic of the recommendation problem. Below we draw some general conclusions that may be useful for selecting an optimal model:

- Meta-User2Vec method gives better results for MovieLens and Deskdrop datasets when the user and item metadata are available. The LightFM method performs significantly better in most settings when only the user and item ids are available (but this approach cannot be applied for the cold-start settings).
- We recommend using the DBOW architecture if the majority of interactions are within the warm-start scenario. However, we observed that for the new-user situation, the PV-DM model with NS exponent-1 performs significantly better. Hence, if, in a given recommendation setting, many new users are expected, this configuration is recommendable.
- The negative sampling exponent parameter may have an important impact on the results, depending on the dataset characteristics. For MovieLens, negative values give better results probably because the dataset is relatively dense and it is better to sample less popular items. However, for Deskdrop and Onet, positive values

are better (the dataset is more sparse than MovieLens and the item metadata are extracted from text or URLs where the size of the dictionary is much larger than the movie genres and the year of release). For the warm-start situation, the size of the window has a significant impact on the DBOW model: the best results were achieved for the largest window size.

- Our analysis of metadata embeddings similarities from the DBOW method with Skip-Gram pre-training shows the latent relations of the user behavioral patterns, but it also reveals some stereotypes and possible data biases. This indicates that the proposed method may be used to increase transparency and identify potential disparities that could be amplified by the recommendation algorithm. Hence, we recommend comparing similarities among the labels for known sensitive attributes (such as the user's gender or nationality) to gather insights into data characteristics.

To summarize, our proposed Meta-User2Vec achieves comparable scores to the LightFM method in the cold-start situations when the user and item metadata are available. However, it is quite sensitive to the choice of hyper-parameters as different settings are optimal for different dataset characteristics, and the default parameters based on NLP applications may not be optimal for a recommendation setting (a similar finding was made by Caselles-Dupré et al. (2018) for the Word2Vec model). Hence, in situations when the data characteristics in the online experiment are unknown, LightFM may give more stable results. Nevertheless, the Meta-User2Vec method may be useful for analyzing the behavioral patterns and relations among the user and the item metadata. Moreover, the resulting embeddings may be incorporated as a pre-training stage to accelerate the training of more complex or hybrid models.

6 Conclusions and future work

We proposed an unsupervised Meta-User2Vec method for building user representations using metadata neural embeddings that may be applied for generating recommendation lists for both the new-user and the item cold-start situations. The proposed approach was tested in five experimental scenarios for warm- and cold-start recommendations on three real-world article datasets.

The results show that after a proper parameter selection, our proposed approach may be used as an alternative for standard matrix factorization-based methods as an initial strategy in cold-start situations, especially when both the user and the item metadata are available.

Additionally, our proposed solution enables representing user and item labels in the same vector space, thereby making it possible to analyze latent behavioral patterns of different groups of users and to detect potential dataset biases. We performed a qualitative analysis of these lists, leading to some interesting insights into the audience behavior, and revealing some latent interest features of distinct user types.

In the future, we plan to incorporate additional user and item metadata as well as contextual features (such as season and time) and combine this knowledge with behavioral data to address the *incomplete cold-start* scenario for session-based recommendations.

Due to the simplicity of its implementation, which is based on the popular Doc2Vec architecture, our proposed method has the potential to be deployed in real-world recommendation settings. Moreover, metadata embeddings may be used as an input for other more complex recommendation approaches to reduce their training time.

References

Alekseev, A., Nikolenko, S.: Word embeddings for user profiling in online social networks. Comput. Sistemas **21**(2), 203–226 (2017)

Arora, S., Liang, Y., Ma, T.: A simple but tough-to-beat baseline for sentence embeddings. In: International Conference on Learning Representations (ICLR 2017), (2016)

Barkan, O., Koenigstein, N.: Item2vec: Neural item embedding for collaborative filtering. CoRR. arXiv:abs/1603.04259, (2016)

Barkan, O., Koenigstein, N., Yogev, E.: The deep journey from content to collaborative filtering. CoRR. arXiv:abs/1611.00384, (2016)

Barkan, O., Koenigstein, N., Yogev, E., Katz, O.: Cb2cf: A neural multiview content-to-collaborative filtering model for completely cold item recommendations. In: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, pp. 228–236, New York, NY, USA, 2019.

Caselles-Dupré, H., Lesaint, F., Royo-Letelier, J.: Word2vec applied to recommendation: Hyperparameters matter. In: Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18, pp. 352–356, New York, NY, USA, 2018.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., Baroni, M.: What you can cram into a single vector: Probing sentence embeddings for linguistic properties. CoRR, arXiv:abs/1805.01070, (2018)

Dacrema, M. F., Cremonesi, P., Jannach, D.: Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, pp. 101–109, New York, NY, USA, 2019.

Dai, A. M., Olah, C., Le, Q. V.: Document embedding with paragraph vectors. CoRR. arXiv:abs/1507.07998, (2015)

de Souza Pereira Moreira, G., Ferreira, F., da Cunha, A. M.: News session-based recommendations using deep neural networks. In: Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems, DLRS 2018, pages 15–23, New York, NY, USA, 2018.

Goel, S., Hofman, J. M., Sirer, M. I.: Who Does What on the Web: A Large-Scale Study of Browsing Behavior. In: ICWSM, (2012)

Grbovic, M.: Listing embeddings in search ranking. https://medium.com/airbnb-engineering/listing-embeddings-for-similar-listing-recommendations-and-real-time-personalization-in-search-601172f7603e, (2018)

Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V., Sharp, D.: E-commerce in your inbox: Product recommendations at scale. CoRR. arXiv:abs/1606.07154, (2016)

Harper, F .M., Konstan, J .A.: The movielens datasets: history and context. ACM Trans. Interact. Intell. Syst. **5**(4), 19:1–19:19 (2015)

Karam, R.: Using word2vec for music recommendations. https://towardsdatascience.com/using-word2vec-for-music-recommendations-bb9649ac2484, (2017)

Kula, M.: Metadata embeddings for user and item cold-start recommendations. CoRR. arXiv:abs/1507.08439, (2015)

Kumar, V., Khattar, D., Gupta, S., Gupta, M., Varma, V.: Deep neural architecture for news recommendation. In: CLEF, (2017)

Lau, J. H., Baldwin, T.: An empirical evaluation of doc2vec with practical insights into document embedding generation. CoRR. arXiv:abs/1607.05368, (2016)

Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning, Vol 32, ICML'14, pages II–1188–II–1196. JMLR. org (2014)

Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. Nature **401**, 788–791 (1999)

McCormick, C.: Applying word2vec to recommenders and advertising. http://mccormickml.com/2018/06/15/applying-word2vec-to-recommenders-and-advertising/, (2018)

McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction, (2018)

Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. CoRR, arXiv:abs/1301.3781, (2013)

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger (editors) Advances in Neural Information Processing Systems 26, pp. 3111–3119. Curran Associates, Inc., (2013)

Misztal-Radecka, J.: Building a semantic user profile for a polish web news portal. Comput. Sci. **19**(3), 307–332 (2018)

Musto, C., Semeraro, G., de Gemmis, M., Lops, P.: Learning word embeddings from wikipedia for content-based recommender systems. In: Ferro, N., Crestani, F., Ferro, N., Crestani, F., Moens, M.-F., Mothe, J., Silvestri, F., Di Nunzio, G .M., Hauff, C., Silvello, G. (eds.) Adv. Inform. Retriev., pp. 729–734. Springer, Cham (2016)

Ozsoy, M. G.: From word embeddings to item recommendation. CoRR. arXiv:abs/1601.01356, (2016)

Phi, V.-T., Chen, L., Hirate, Y.: Distributed representation-based recommender systems in e-commerce. In: DEIM Forum, (2016)

Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp 45–50, Valletta, Malta, (May 2010). ELRA. http://is.muni.cz/publication/884893/en

Stiebellehner, S., Wang, J., Yuan, S.: Learning continuous user representations through hybrid filtering with doc2vec. CoRR. arXiv:abs/1801.00215, (2018)

Vasile, F., Smirnova, E., Conneau, A.: Meta-prod2vec: Product embeddings using side-information for recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16, pp 225–232, New York, NY, USA, 2016.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S. R.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. CoRR. arXiv:abs/1804.07461, (2018)

Wang, C., Blei, D. M.: Collaborative topic modeling for recommending scientific articles. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, pages 448–456, New York, NY, USA, (2011).

Wang, Y., Wang, L., Li, Y., He, D., Liu, T., Chen, W.: A theoretical analysis of NDCG type ranking measures. CoRR. arXiv:abs/1304.6480, (2013)

Wei, J., He, J., Chen, K., Zhou, Y., Tang, Z.: Collaborative filtering and deep learning based recommendation system for cold start items. Expert Syst. Appl. **69**, 10 (2016)

Xu, Q., Erman, J., Gerber, A., Mao, Z., Pang, J., Venkataraman, S.: Identifying diverse usage behaviors of smartphone apps. In: Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, IMC '11, pp 329–344, New York, NY, USA, 2011.

Zheng, L., Noroozi, V., Yu, P. S.: Joint deep modeling of users and items using reviews for recommendation. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17, pp 425–434, New York, NY, USA, (2017).

Zhou, Y., Nadaf, A.: Embedded collaborative filtering for "cold start" prediction. CoRR. arXiv :abs/1704.02552, (2017)

Joanna Misztal-Radecka is a Ph.D. candidate at AGH University of Science and Technology and a Senior Data Scientist at Ringier Axel Springer Polska. Her main research interests are recommendation systems, natural language processing, and machine learning explainability. Apart from the research, she has worked on many industrial applications of machine learning, in particular the large-scale news recommendation system for media services of Ringier Axel Springer.

Bipin Indurkhya is a professor of Cognitive Science at the Jagiellonian University, Krakow, Poland. His main research interests are social robotics, usability engineering, affective computing and creativity. He received his Master's degree in Electronics Engineering from the Philips International Institute, Eindhoven (The Netherlands) in 1981, and PhD in Computer Science from University of Massachusetts at Amherst in 1985. He has taught at various universities in the US, Japan, India, Germany and Poland; and has led national and international research projects with collaborations from companies like Xerox and Samsung.

Dr. Aleksander Smywiński-Pohl is a researcher in natural language processing. He received his Ph.D. in 2015 from AGH University of Science and Technology in Krakow for the work entitled: Automatic extraction of semantic relations from Polish texts. His primary research interests concentrate on the application of modern NLP techniques in a broad range of practical problems. In 2017 he started a research project funded by Polish National Center for Research and Development devoted to the construction of an intelligent legal information system called „Lemkin". He also participated in other projects aimed at building the Polish language model for application in Automatic Speech Recognition, sentiment analysis of user-generated content, monitoring contents of the public media as well as cyberbullying and self-harm detection.