Reviews Are Gold!? On the Link between Item Reviews and Item Preferences

Tobias Eichinger[1]

[1]Technical University of Berlin, Straße des 17. Juni 135, Berlin, 10623, Germany

Abstract

User-user similarities in recommender systems are traditionally assessed on co-rated items. As ratings encode item preferences, similarities on co-rated items capture similarities in item preferences. However, a majority of similarities are undefined as particularly small profiles seldom overlap. We propose to use a similarity measure based on users' item reviews in order to estimate similarities in item preferences in the absence of co-rated items. Although it is commonly believed that item reviews are descriptive of a user's item preferences, it is not clear whether indeed and what about a user's item preferences item reviews describe. We present empirical results indicating that the proposed review-based similarity measure captures features in users' item preferences that are different from those captured on co-rated items. Astonishingly, we find that 10 keywords of a user's item reviews suffice to represent a user's item preferences. Independently, we argue that the proposed review-based similarity measure is particularly suitable for use in decentralized recommender systems for three design properties. First, it can be calculated between any pair of users who hold item reviews. Second, it can be calculated bilaterally without involvement of a third party. And third, it does not require to reveal a user's plain review text.

Keywords

review-based similarity, word mover's distance, word embedding, fasttext, keyword extraction, YAKE,

1. Introduction

We denote by *scarcity* the situation that only a small subset of rows in the user-item matrix are available for recommendation. Scarcity is commonly encountered in decentralized recommender systems in which users only have access to a small subset of other users. Scarcity is often considered beneficial for user privacy [1, 2], yet detrimental to recommendation performance. Sharing rating profiles in order to alleviate scarcity is problematic as rating profiles are often considered personal data, and sensitive as such. A compromise is commonly made by only sharing ratings with similar users, where similarity is measured with respect to item preference. As similarities are traditionally calculated on users' item ratings, it is not trivial to find users with similar item preferences without sharing one's item ratings.

Approximate and exact methods have been proposed to calculate the similarity between users on ratings without revealing them. Lathia [3] proposes an approximate similarity measure that does not require to disclose neither the rated items nor their actual ratings. Other approximate methods include profile obfuscation [4, 5, 6]. Exact similarity estimates can be obtained through cryptographic methods bilaterally [7, 8], or with the help of a third-party [9]. Despite the feasibility to compute similarities in a privacy-preserving fashion, the problem remains that similarity measures usually require that some items are rated by both users. Such items are typically denoted *co-rated* items. As user-item matrices are commonly very sparse, similarity measures based on co-rated items are undefined for a majority of user-user pairs. This circumstance is exacerbated under scarsity.

We propose to calculate similarities between users on the basis of their item reviews instead of co-rated items. We particularly address scenarios in which sparsity meets scarcity. Reviews are commonly believed to be descriptive of a user's item preferences. This belief is supported by reports on the success of state-of-the-art-algorithms that leverage reviews [10, 11]. However, a recent review by Sachdeva and McAuley [12] puts this belief into question. Their findings indicate that state-of-the-art algorithms that leverage item reviews do not consistently outperform even simple baselines that do not. This inconsistency raises the question whether state-of-the-art methods are in fact able to reliably extract information from reviews that is beneficial for recommendation. Reviews remain complex inputs to recommendation algorithms that seem to defy current endeavors to extract users' item preferences reliably.

In order to better understand what about users' item preferences is reflected in reviews, we propose a similarity measure that compares users on the basis of their item reviews. We report findings on our pilot experi-

3rd Edition of Knowledge-aware and Conversational Recommender Systems (KaRS) & 5th Edition of Recommendation in Complex Environments (ComplexRec) Joint Workshop @ RecSys 2021, September 27–1 October 2021, Amsterdam, Netherlands
tobias.eichinger@tu-berlin.de (T. Eichinger)
https://www.snet.tu-berlin.de/menue/team/tobias_eichinger/ (T. Eichinger)
0000-0002-8351-2823 (T. Eichinger)

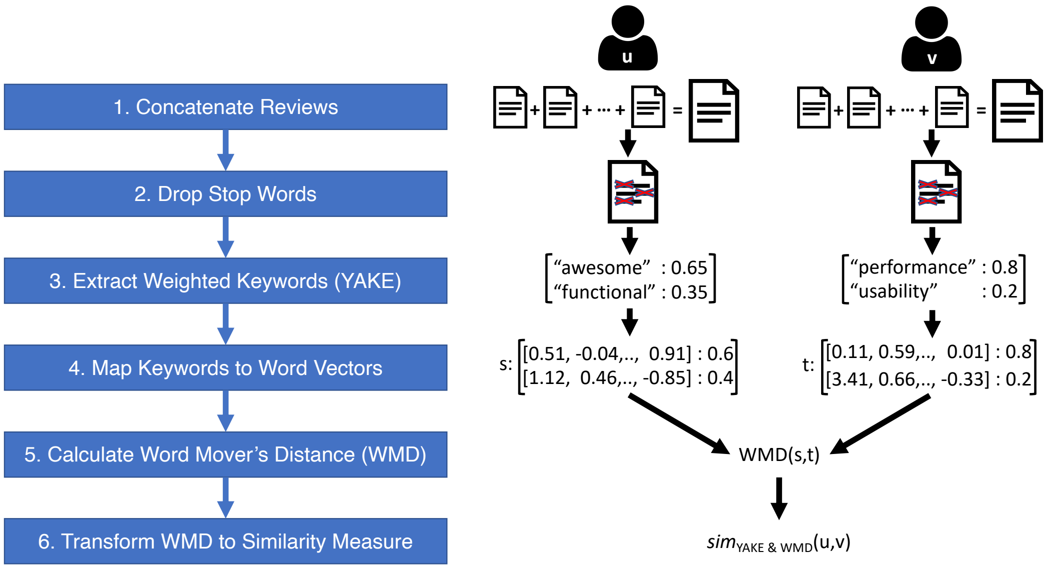

Figure 1: Similarity comparison between users u and v on the basis of their item reviews. The comparison procedure follows a six-step approach based on [13].

ments indicating that the proposed similarity measure (a) indeed captures similarity in users' item preferences, and (b) captures features that are different from those captured by co-rated-items-based similarity measures.

Independently from the above results, we find that the design of the proposed review-based similarity measure motivates its use in decentralized recommender systems for three design properties. First, it can be calculated between any pair of users who hold item reviews. Second, it can be calculated bilaterally without involvement of a third party. And third, it does not require to reveal a user's plain review text.

2. Concept

We follow along the lines of the user-user similarity measure proposed by Eichinger et al. [13]. It has originally been proposed as a general-purpose similarity measure on texting data. In the paper at hand, we instead apply it to item reviews and show that it particularly captures similarity in users' item preferences.

Similarity comparison can be summarized as a six-step approach as shown in Figure 1. We first elaborate on Steps 4.-6. in Section 2.1, which constitute the core of the similarity measure. Afterwards in Section 2.2, we focus on optional steps such as text preprocessing and keyword extraction comprising Steps 1.-3. Eichinger et al. originally proposed keyword extraction on the basis of tf-idf features. In contrast to the original work, we instead apply a state-of-the-art keyword extractor, which additionally allows users to run keyword extraction independently from other users.

2.1. From Document Distance to Review Similarity

4. Map Keywords to Word Vectors (Figure 1): Kusner et al. [14] propose the Word Mover's Distance (WMD), a distance metric between text documents that are each represented by a subset of their words.[1] The WMD is made such that text documents that hold semantically similar words – and thus not necessarily the same words – are close. Semantic similarity between words is captured by word embeddings. Word embeddings map words to word vectors such that word vectors of semantically similar words are close. Words need not necessarily be keywords. Note that all users need to use the same word embedding model, wherefore we use a publicly available pre-trained word embedding model.

5. Calculate Word Mover's Distance (Figure 1): The WMD leverages word vectors that condense semantic similarity

[1]The WMD is more broadly known as Earth Mover's Distance (EMD), where the EMD is in turn a special Wasserstein metric.

between single words, in order to measure semantic similarity between sets of words. More precisely, the WMD compares so-called *signatures*.[2] Signatures are sets of word vectors in which every word vector is associated with a word weight. The number of word vectors in a signature is called the signature size. The distance between two signatures, associated with the distance between two text documents, is then determined by solving a transportation problem (see [14] for details). The WMD can be calculated bilaterally and independently of other users upon the exchange of signatures.

6. Transform WMD to Similarity Measure (Figure 1): We transform the WMD distance metric into a similarity measure. Note that the WMD distance between two similar text documents is close to zero, whereas dissimilar text documents may yield arbitrarily large WMD distances. Hence, we first limit the co-domain to $\mathrm{WMD}(s,t) \in [0,2]$ for any pair of signatures s and t. We do so by using the cosine distance[3] to measure distances between word vectors and normalize word weights in a signature such that they sum to 1.[4] We then obtain a similarity measure upon the following linear transformation:

$$sim_{\mathrm{WMD}}(s,t) := 1 - \frac{1}{2}\mathrm{WMD}(s,t) \in [0,1].$$

The signature size is the sole hyperparameter of the WMD, and thus also of the associated similarity measure sim_{WMD}. We will specify the signature size where required, yet omit it in the notation for reasons of brevity.

2.2. Key Word Extraction

1. Concatenate Reviews (Figure 1): In order to arrive at a user-specific text document, we first concatenate all item reviews authored by a user in arbitrary order with blanks between reviews.

2. Drop Stop Words (Figure 1): We drop stop words as a basic text preprocessing step. We do not perform any further preprocessing in order to mitigate the impact due to preprocessing on the evaluation of the proposed review-based similarity measure.

3. Extract Weighted Keywords (Figure 1): The computational complexity of the WMD is often prohibitive as it is supercubic in the signature size. For this reason, the WMD has not found wide adoption. Efforts to lower the computational complexity include approximation [16, 17] and the reduction of the signature size by keyword extraction [13]. In a previous paper, we applied keyword extraction on the basis of the tf-idf word relevance measure [13]. Note that keyword extraction via tf-idf requires to keep track of the global usage of terms in all users' reviews. A more convenient alternative is Yet Another Keyword Extractor (YAKE) by Campos et al. [18, 19]. Their keyword extractor is document-based and works on textual features of single documents. It does not require information on other documents.

YAKE is a weighted keyword extractor.[5] It attaches positive keyword weights $g_i > 0$ to every keyword w_i of a text document. Keywords in YAKE are considered more important in describing their underlying text document the smaller their associated keyword weights are. Conversely, WMD word weights are considered more important the larger they are. We therefore reverse the order of the keyword weights g_i for use as word weights in the WMD. We do so via the linear transformation defined by $\underline{g_i} := g_{max} + g_{min} - g_i \in [g_{min}, g_{max}]$, and consecutive normalization of the word weights such that they sum to 1, where g_{min} and g_{max} are the minimum and maximum keyword weights respectively.

Applying YAKE in conjunction with the above weight transformation on a user's item reviews yields signatures that serve as input to the WMD. We denote by $\underline{\mathrm{YAKE}}(u)$ the thus associated signature of some user u's item reviews. Combining this with the results of Section 2.1, we can now define the review-based similarity measure as

$$sim_{\mathrm{review}}(u,v) := sim_{\mathrm{WMD}}(\underline{\mathrm{YAKE}}(u), \underline{\mathrm{YAKE}}(v)),$$

where u and v are some users that hold item reviews.

3. Evaluation

We present results that contrast the cosine similarity as a traditional co-rated-items-based similarity measure with the proposed review-based similarity measure sim_{review}. We emphasize that the goal we pursue by this comparison is not to argue that the review-based similarity measure is superior to co-rated-items-based similarity measures. Instead, it is our goal to find a solid indication that the review-based similarity measure indeed captures similarity in terms of item preferences between users. We employ the rationale, that if the review-based similarity measure captures similarity

[2] The term *signature* has been coined by Rubner et al. [15] in the domain of computer vision as abstractions of color histograms.

[3] $d_{\cos}(u,v) = 1 - \frac{<u,v>}{\|u\| \cdot \|v\|}$, where $< \cdot , \cdot >$ denotes the dot product and $\|\cdot\|$ the Euclidean norm.

[4] If the Euclidean distance is preferred, we can alternatively normalize vectors to length 1 and normalize word weights such that they sum to 1.

[5] YAKE also extracts keyphrases. However, we only consider keywords and omit treatment of keyphrases for reasons of simplicity. Although it is also possible to convert keyphrases into vectors, distinct scientific reasoning is required to justify a comparison between signatures that combine word vectors and phrase vectors.

Table 1

Descriptive Statistics of the data sets High-100, Median-100, Low-100, and Mix-100. *: Signature size between 1 and 100 that produces the best RMSE on user-based CF as per Equation (1). Tie-breakers are resolved by choice of the smallest signature size. **: Percentage of user pairs that have co-rated items in the training set.

	Head-100	Median-100	Tail-100	Mix-100
unique users	100	100	100	100
unique items	180, 981	797	499	70, 213
ratings/reviews	278, 927	800	500	83, 714
optimal signature size*	5	–	–	10
pairs with co-rated items**	81.82%	0.03%	0.01%	8.77%

between users' item preferences, then it necessarily must perform well in user-based Collaborative Filtering (CF).

We calculate sim_{review} with the help of the following software contributions.[6] We use Pele and Werman's Python implementation of the WMD [20].[7] We use Bojanowski et al.'s publicly available pre-trained *fasttext* word embedding model *cc.en.300.bin* [21].[8] We use the stop word list provided by Bird's The Natural Language Toolkit (NLTK) [22]. Finally, we use Campos et al.'s Python library *yake* [18].[9]

Splits into training and test sets are at a ratio of 80 to 20, where particularly every user's entries are split into portions of training and test entries. We report average results over 5 distinct training-test splits on the usual Root Mean Squared Error (RMSE) accuracy metric.

3.1. Data Sets

We present results on two small samples of the *Amazon Reviews* 5*-core (2014)* data set [23].[10] The original data set holds roughly 41 million entries on 24 product domains, where every user has at least 5 rating-review pairs. The two samples considered in the paper at hand cover two distinct scenarios of (a) an artificially high and (b) a more realistic density. We now describe their construction.

We draw the first data set **Head-100** by selecting the 100 largest user profiles. It simulates an artificially high density of ratings and reviews with particularly large amounts of review text per user. As for the second data set **Mix-100**, we draw 2 additional data sets Median-100 and Tail-100, of medium and low density, by selecting the profiles of 100 median and 100 tail users respectively. We finally sample Mix-100 from the datasets Head-100, Median-100, and Tail-100 at a ratio of 33 to 34 to 33. We construct Mix-100 in this way in order to guarantee the presence of large, medium-sized, and small profiles in the sample. Some descriptive statistics are shown in Table 1. Note that, if similarity is measured on the basis of co-rated items, only 0.03% and 0.01% of all pairwise similarites can be calculated for users in the Median-100 and Low-100 data sets respectively. As we compare review-based with co-rated-items-based similarity measures, we omit an analysis on the samples Median-100 and Low-100 as they simply provide too little ground for comparison.

3.2. Baselines

We apply the following standard mean-centered rating estimation equation for user-based CF:

$$\hat{r}_{u,i} = \bar{r}_u + \sum_{v \in N_{u,i}} \frac{sim(u,v)(r_{v,i} - \bar{r}_v)}{\sum_{v \in N_{u,i}} sim(u,v)}, \qquad (1)$$

where $\hat{r}_{u,i}$ denotes an estimated rating for user u on item i, $\bar{r}_u$ the mean rating of user u, $N_{u,i}$ some neighborhood of users of user u that have rated item i, sim a user-user similarity measure, and $r_{v,i}$ the true rating of user v on item i. For reasons of brevity we say that a similarity measure outperforms another, when in fact we mean that rating estimation as per Equation (1) equipped with the one similarity measure outperforms that equipped with the other.

We propose two similarity measures as baselines for comparison with sim_{review}. First, cosine similarity as a similarity measure based on mutually rated items. And second, a simple arithmetic mean having equal similarity weights $sim_{\text{mean}}(u,v) = 1/|N_{u,i}|$ for all users $v \in N_{u,i}$. If sim_{review} outperforms sim_{mean}, it is an indication that sim_{review} does capture similarity in item preference between users, that is more than an estimate without prior knowledge on reviews. If further sim_{review} outperforms sim_{cosine}, it is an indication that the review-based similarity measure captures similarity in item preference at least on a par with co-rated-items-based similarity measures.

[6] https://github.com/TEichinger/WMDtestbed
[7] https://pypi.org/project/pyemd/
[8] https://fasttext.cc/docs/en/pretrained-vectors.html
[9] https://pypi.org/project/yake/
[10] https://doi.org/10.7910/DVN/V7X3VE

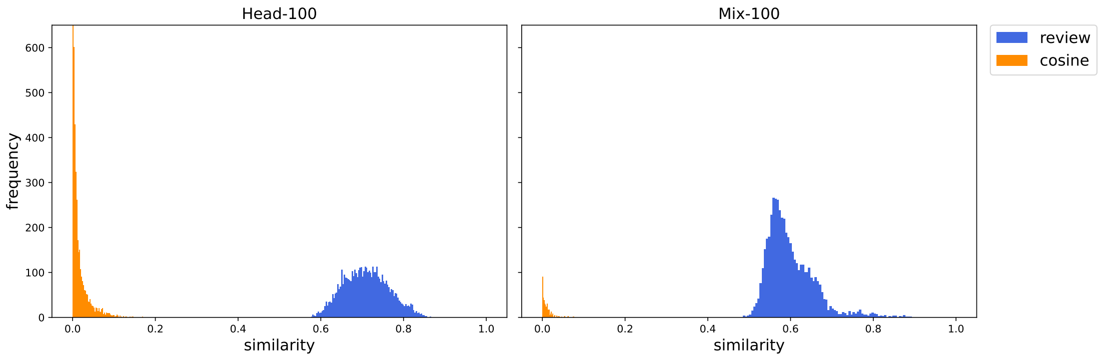

Figure 2: Comparison of histograms of pairwise similarities for the review-based similarity measure sim_{review} and the cosine similarity sim_{cosine} for the training sets Head-100 and Mix-100.

3.3. Capture Similarity in Item Preference on Item Reviews

We present findings that indicate that the review-based similarity measure sim_{review} captures similarity in item preference. We first contrast the statistical properties of sim_{review} and sim_{cosine}, where we assume that sim_{cosine} already captures some similarity in item preference. We then measure their respective impact on rating estimation as per Equation (1), acting in the role of (a) similarity weights, and (b) a neighborhood selection criterion. In order to study the impact due to (a) and (b) individually, we first omit neighborhood selection by setting $N_{u,i}$ as the set of all other users $v \neq u$ and applying sim_{review} similarity weights, and then conversely omit similarity-based weighted averaging by using sim_{mean} similarity weights and setting $N_{u,i}$ as the set of k users that are most similar to user u with respect to sim_{review} and have rated item i.

3.3.1. Statistical Properties

The similarity measures sim_{review} and sim_{cosine} capture distinct aspects of similarity between users' item preferences. We find that sim_{review} and sim_{cosine} are only weakly positively correlated with respect to the Spearman rank correlation (0.36 on High-100, and 0.25 on Mix-100). If in contrast both similarity measures were strongly positively correlated, this would indicate that both similarity measures capture similar aspects of similarity. In that case, we would also expect that both similarity measures yield similar recommendation performance.

We further find that sim_{review} and sim_{cosine} produce distinct similarity distributions. Figure 2 shows the similarity distributions in the data sets High-100 and Mix-100. We observe that the cosine similarity's distribution follows the shape of an exponential distribution in both High-100 and Mix-100. In contrast, sim_{review}'s distribution follows the shape of a normal distribution in High-100, and a mix of normal distributions with distinct modes in Mix-100. Review-based similarities in High-100, Median-100, and Low-100 seem to have distinct modes that interfere in Mix-100.

Table 2

Recommendation accuracy (RMSE) over various minimum neighborhood sizes $n_{\min} = \min_{u,i} |N_{u,i}|$ on the Head-100 and Mix-100 data sets. Bold values are per-column best values, where asterisks indicate statistical significance of paired t-tests to the alternatives ($\alpha = 0.05$). Hyphens indicate that no ratings could be estimated due to insufficiently small neighborhood size. We see that weighing ratings via sim_{review} in Equation (1) consistently outperforms the baselines for $n_{\min} \leq 5$ on average.

Head-100

$n_{\min}$	1	2	3	5	10
sim_{review}	**0.962**	**0.911**	**0.899**	**0.903**	0.880
sim_{cosine}	0.968	0.923	0.905	0.904	**0.877**
sim_{mean}	0.962	0.912	0.900	0.903	0.880

Mix-100

$n_{\min}$	1	2	3	5	10
sim_{review}	**1.084**	**1.019***	**0.985***	**0.905**	–
sim_{cosine}	1.087	1.033	0.997	0.932	–
sim_{mean}	1.085	1.022	0.988	0.988	–

3.3.2. Similarity-based Weighted Averaging

We find that sim_{review} similarity weights provide significantly better recommendation performance, if profiles are not exclusively large such as in Mix-100. If in con-

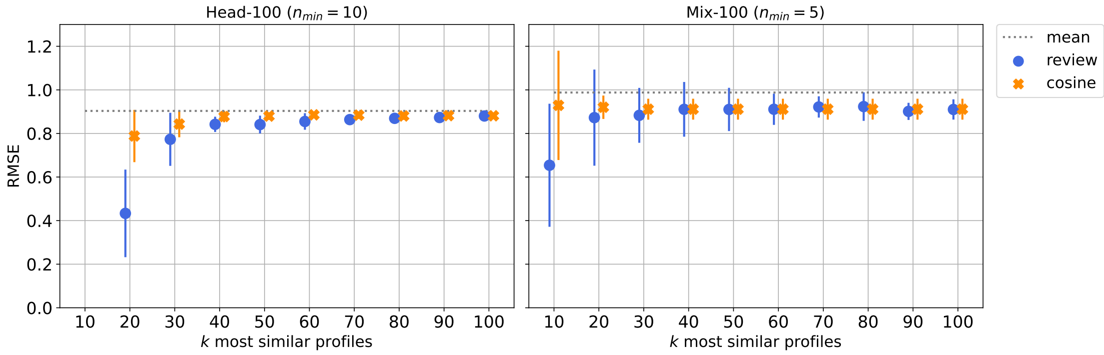

Figure 3: Recommendation accuracy (RMSE) on the k most similar profiles on the Head-100 and Mix-100 data sets on sim_{mean} similarity weights. Error bars show $\pm$ one standard deviation. For $k = 10$ in Head-100, no ratings could be estimated due to a lack of ratings.

trast, profiles are exclusively large such as in Head-100, sim_{review} similarity weights, sim_{cosine} similarity weights, and sim_{mean} similarity weights perform similarly as shown in Table 2. None performs significantly better on Head-100. On Mix-100 however, sim_{review} significantly outperforms the alternatives for $n_{\min} \in \{2, 3\}$. For $n_{\min} = 5$, superiority is not statistically significant despite the large absolute margin due to sim_{review}'s high empirical standard deviation.

3.3.3. Similarity-based Profile Selection

We find that performing rating estimation on only the k most similar user profiles based on sim_{review} outperforms both sim_{cosine} and sim_{mean} on average. More concretely, we see in Figure 3 that sim_{review} and sim_{cosine} perform similarly for $k \geq 40$ on both Head-100 and Mix-100. For $k \leq 30$, we see that decreasing k simulatenously yield decreasing RMSE values on Head-100. On Mix-100, only the RMSE of sim_{review} decreases for decreasing k, while the RMSE of sim_{cosine} essentially stays the same. This is due to the fact that many pairwise sim_{cosine} values are undefined such that increasing k does not yield larger neighborhoods $N_{u,i}$. We observe further that RMSE mean values tend to decrease with decreasing parameter values k, whereas RMSE standard deviations increases with decreasing parameter values k.

4. Related Work

We find related work on the following three aspects. First, leveraging review text for recommendation in general. Second, estimating similarity in item preference without using ratings. And third, alternatives to the proposed YAKE keyword extractor and *fasttext* word embedding models for use in sim_{review}.

Item Reviews for Item Recommendation: There is a wealth of work that aims to leverage item reviews in order to improve recommendation performance. Sachdeva and McAuley [12] recently presented a review of state-of-the-art recommender algorithms that leverage review data. They categorize them into two tracks. First, algorithms that use reviews for regularization at algorithm training time [24, 25]. And second, algorithms that use review-based features for use at recommendation time [26, 10, 11, 25, 27, 28, 29]. In the paper at hand, we propose a review-based similarity measure as a feature that captures similarity in users' item preferences. We thus contribute to the second category.

Estimating Similarity in Item Preference without Using Ratings: Similarity in item preference can for instance be estimated on the basis of the shared context of users. Wainakh et al. [2] show that users who share a social context also tend to share item preferences. More precisely, they show that profiles sampled from users close in the social graph provide better recommendation accuracy on an association rules mining algorithm as compared to uniformly randomly sampled profiles. de Spindler et al. [30] propose to use geo-temporal context between users as a proxy to elicit mutual item preferences in opportunistic networking scenarios.

Alternative Keyword Extractors and Word Embeddings: The literature proposes a large spectrum of keyword extractors and word embedding models. We apply YAKE as a state-of-the-art keyword extractor [18, 19]. It runs on single documents rather than a corpus of documents. Keyword extraction can thus be performed by users individually. An alternative that also runs on single documents is RAKE [31]. A majority of keyword extractors require a document corpus for keyword extraction [32, 33, 34].

We apply a *fasttext* word embedding model since it can map word tokens that have not been seen at training time by leveraging subword information [21]. An alternative that also leverages subword information is for instance *LexVec* [35]. A majority of word embedding models does not leverage subword information and can thus only map word tokens available at training time [36, 37, 38, 39].

5. Conclusion

We propose a review-based user-user similarity measure that presents an alternative to traditional co-rated-items-based similarity measures. It is particularly suitable, if two users do not have co-rated items and would thus default to an undefined user-user similarity. Similarities can now be calculated on the basis of item reviews instead of co-rated items.

We find that the proposed review-based similarity measure captures similarity in users' item preferences. Interestingly, the proposed review-based similarity measure captures different features from those captured on co-rated items. The difference can be linked implicitly to the difference in their statistical features such as similarity distribution and Spearman rank correlation. However, the difference cannot be characterized explicitly as the review-based similarity measure is based on unsupervised word embeddings. More precisely, word embeddings find semantic similarity between words, yet do not tell how and in which sense the words are similar.

We conclude that the proposed review-based user-user similarity measure presents a promising feature for recommender system design, when item reviews are available. We do not argue that the proposed review-based similarity measure is in any sense superior to co-rated-items-based similarity measures. On the contrary, our findings indicate that they are complementary in modeling users' item preferences.

References

[1] M. Larson, A. Zito, B. Loni, P. Cremonesi, Towards minimal necessary data: The case for analyzing training data requirements of recommender algorithms, in: Proceedings of the 1st FATREC Workshop @ RecSys, 2017, pp. 1–6. doi:10.18122/B2VX12.

[2] A. Wainakh, T. Grube, J. Daubert, M. Mühlhäuser, Efficient privacy-preserving recommendations based on social graphs, in: Proceedings of the 13th ACM Conference on Recommender Systems, ACM, 2019, pp. 78–86. doi:10.1145/3298689.3347013.

[3] N. Lathia, S. Hailes, L. Capra, Private distributed collaborative filtering using estimated concordance measures, in: Proceedings of the 1st ACM Conference on Recommender Systems, ACM, 2007, pp. 1–8. doi:10.1145/1297231.1297233.

[4] S. Berkovsky, T. Kuflik, F. Ricci, The impact of data obfuscation on the accuracy of collaborative filtering, Expert Systems with Applications 39 (2012) 5033–5042. doi:10.1016/j.eswa.2011.11.037.

[5] R. Parameswaran, D. M. Blough, Privacy preserving collaborative filtering using data obfuscation, in: Proceedings of the 3rd IEEE International Conference on Granular Computing, 2007, pp. 380–380. doi:10.1109/GrC.2007.133.

[6] H. Polat, W. Du, Privacy-preserving collaborative filtering using randomized perturbation techniques, in: Proceedings of the 3rd IEEE International Conference on Data Mining, 2003, pp. 625–628. doi:10.1109/ICDM.2003.1250993.

[7] M. Alaggan, S. Gambs, A.-M. Kermarrec, Private similarity computation in distributed systems: from cryptography to differential privacy, in: OPODIS, volume 7109 of *LNCS*, Springer Berlin Heidelberg, 2011, pp. 357–377. doi:10.1007/978-3-642-25873-2_25.

[8] D. Yang, C. Lin, B. Yang, A novel secure cosine similarity computation scheme with malicious adversaries, International Journal of Network Security & Its Applications 5 (2013) 171–178. doi:10.5281/zenodo.4032143.

[9] J. Zhang, S. Hu, Z. L. Jiang, Privacy-preserving similarity computation in cloud-based mobile social networks, IEEE Access 8 (2020) 111889–111898. doi:10.1109/ACCESS.2020.3003373.

[10] F. Chen, Z. Dong, Z. Li, X. He, Federated meta-learning for recommendation, 2018. URL: http://arxiv.org/abs/1802.07876.

[11] Y. Tay, A. T. Luu, S. C. Hui, Multi-pointer co-attention networks for recommendation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 2309–2318. doi:10.1145/3219819.3220086.

[12] N. Sachdeva, J. McAuley, How useful are reviews for recommendation? a critical review and potential improvements, in: Proceedings of the 43rd International ACM SIGIR Conf. on R&D in Information Retrieval, ACM, 2020, pp. 1845–1848. doi:10.1145/3397271.3401281.

[13] T. Eichinger, F. Beierle, S. U. Khan, R. Middelanis, V. Sekar, S. Tabibzadeh, affinity: A system for latent user similarity comparison on texting data, in: Proceedings of the 53rd IEEE Int. Conf. on Comm., IEEE, 2019, pp. 1–7. doi:10.1109/ICC.2019.8761051.

[14] M. Kusner, Y. Sun, N. Kolkin, K. Weinberger, From word embeddings to document distances, in: Pro-

ceedings of the 32nd Int. Conf. on Machine Learning, volume 37, JMLR.org, 2015, pp. 957–966. doi:10.5555/3045118.3045221.
[15] Y. Rubner, C. Tomasi, L. Guibas, A metric for distributions with applications to image databases, in: Proceedings of the 6th IEEE International Conference on Computer Vision, IEEE, 1998, pp. 59–66. doi:10.1109/ICCV.1998.710701.
[16] M. Cuturi, Sinkhorn distances: Lightspeed computation of optimal transport, Advances in Neural Information Processing Systems 26 (2013) 2292–2300. URL: https://papers.nips.cc/paper/2013/hash/af21d0c97db2e27e13572cbf59eb343d-Abstract.html.
[17] K. Atasu, T. Mittelholzer, Linear-complexity data-parallel earth mover's distance approximations, in: Proceedings of the 36th International Conference on Machine Learning, volume 97, PMLR, 2019, pp. 364–373. URL: http://proceedings.mlr.press/v97/atasu19a.html.
[18] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, A. Jatowt, Yake! keyword extraction from single documents using multiple local features, Information Sciences 509 (2020) 257–289. doi:10.1016/j.ins.2019.09.013.
[19] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, A. Jatowt, Yake! collection-independent automatic keyword extractor, in: Advances in Information Retrieval, volume 10772 of *LNCS*, Springer, 2018, pp. 806–810. doi:10.1007/978-3-319-76941-7_80.
[20] O. Pele, M. Werman, Fast and robust earth mover's distances, in: Proceedings of the 12th IEEE International Conference on Computer Vision, IEEE, 2009, pp. 460–467. doi:10.1109/ICCV.2009.5459199.
[21] P. Bojanowski, E. Grave, A. Joulin, T. lov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017) 135–146. doi:10.1162/tacl_a_00051.
[22] S. Bird, NLTK: The Natural Language Toolkit, in: Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions, ACL, 2006, pp. 69–72. doi:10.3115/1225403.1225421.
[23] J. McAuley, C. Targett, Q. Shi, A. van den Hengel, Image-based recommendations on styles and substitutes, in: Proceedings of the 38th International ACM SIGIR Conf. on R&D in Information Retrieval, ACM, 2015, pp. 43–52. doi:10.1145/2766462.2767755.
[24] J. McAuley, J. Leskovec, Hidden factors and hidden topics: Understanding rating dimensions with review text, in: Proceedings of the 7th ACM Conference on Recommender Systems, ACM, 2013, pp. 165–172. doi:10.1145/2507157.2507163.
[25] R. Catherine, W. Cohen, Transnets: Learning to transform for recommendation, in: Proceedings of the 11th ACM Conference on Recommender Systems, ACM, 2017, pp. 288 – 296. doi:10.1145/3109859.3109878.
[26] L. Zheng, V. Noroozi, P. S. Yu, Joint deep modeling of users and items using reviews for recommendation, in: Proceedings of the 10th ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 425–434. doi:10.1145/3018661.3018665.
[27] Y. Bao, H. Fang, J. Zhang, Topicmf: Simultaneously exploiting ratings and reviews for recommendation, in: Proceedings of the 28th AAAI Conference on Artificial Intelligence, AAAI Press, 2014, pp. 2–8. doi:10.5555/2893873.2893874.
[28] K. Bauman, A. Tuzhilin, Discovering contextual information from user reviews for recommendation purposes., in: Proceedings of the 1st CBRecSys Workshop @RecSys, CEUR-WS, 2014, pp. 2–9. URL: http://ceur-ws.org/Vol-1245/cbrecsys2014-paper01.pdf.
[29] P. G. Campos, N. Rodríguez-Artigot, I. Cantador, Extracting context data from user reviews for recommendation: A linked data approach, in: In Proceedings of the 1st ComplexRec Workshop @RecSys, CEUR-WS, 2017, pp. 14–18. URL: http://ceur-ws.org/Vol-1892/paper3.pdf.
[30] A. De Spindler, M. C. Norrie, M. Grossniklaus, Collaborative filtering based on opportunistic information sharing in mobile ad-hoc networks, in: Proceedings of the 2007 OTM Conf. Int. Conf., Springer Berlin Heidelberg, 2007, pp. 408–416. doi:10.5555/1784607.1784643.
[31] S. Rose, D. Engel, N. Cramer, W. Cowley, Automatic keyword extraction from individual documents, in: Text Mining, John Wiley & Sons Ltd., 2010, pp. 1–20. doi:10.1002/9780470689646.ch1.
[32] S. R. El-Beltagy, A. Rafea, Kp-miner: A keyphrase extraction system for english and arabic documents, Information Systems 34 (2009) 132–144. doi:10.1016/j.is.2008.05.002.
[33] R. Mihalcea, P. Tarau, TextRank: Bringing order into text, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, ACL, 2004, pp. 404–411. URL: https://aclanthology.org/W04-3252.
[34] A. Bougouin, F. Boudin, B. Daille, TopicRank: Graph-based topic ranking for keyphrase extraction, in: Proceedings of the 6th International Joint Conference on Natural Language Processing, AFNLP, 2013, pp. 543–551. URL: https://aclanthology.org/I13-1062.
[35] A. Salle, A. Villavicencio, Incorporating subword information into matrix factorization word embed-

dings, in: Proceedings of the 2nd Workshop on Subword/Character Level Models, ACL, 2018, pp. 66–71. doi:10.18653/v1/W18-1209.
[36] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, in: Workshop Track Proceedings of the 1st International Conference on Learning Representations, 2013. URL: https://arxiv.org/abs/1301.3781.
[37] J. Pennington, R. Socher, C. D. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, ACL, 2014, pp. 1532–1543. doi:10.3115/v1/D14-1162.
[38] O. Levy, Y. Goldberg, Linguistic regularities in sparse and explicit word representations, in: Proceedings of the 18th Conference on Computational Natural Language Learning, ACL, 2014, pp. 171–180. doi:10.3115/v1/W14-1618.
[39] M. Nickel, D. Kiela, Poincaré embeddings for learning hierarchical representations, Advances in Neural Information Processing Systems 30 (2017) 6338–6347. URL: https://proceedings.neurips.cc/paper/2017/hash/59dfa2df42d9e3d41f5b02bfc32229dd-Abstract.html.